

Modeling the Spreading of Diseases

Joakim Sundnes^{1,2}

Hans Petter Langtangen^{1,2}

¹Simula Research Laboratory

²University of Oslo, Dept. of Informatics

Nov 8, 2019



0.1 Plan for week 46-47

Monday 11/11:

- Comments on weekly assignments
- Exercise E.29, E.49
- Disease modeling

Thursday 14/11:

- No lecture (probably)

Week 47:

- Monday 18/11: Project lecture/questions (here in Sophus Lie)
- Thursday: No lecture
- Thursday/Friday: Project help ("orakel"). (time and place TBD)

0.2 Modeling a complex phenomenon by simple math (1)

Plan:

- Use simple intuition to derive a system of *difference equations* to model the spread of diseases
- Program the difference equations in the usual way (i.e. `for`-loops)
- Explore possible model extensions
- See how the difference equations correspond to *ordinary differential equations*

0.3 Modeling a complex phenomenon by simple math (2)

Assumptions:

- We consider a perfectly mixed population in a confined area
- No spatial transport, just temporal evolution
- We do not consider individuals, just a grand mix of them

We consider very simple models, but these can be extended to full models that are used world-wide by health authorities. Typical diseases modeled are flu, measles, swine flu, HIV, ...

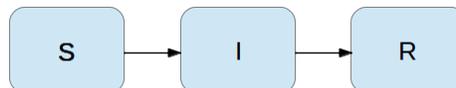
Additional material is available from <https://github.com/hplgit/disease-modeling>.

0.4 We keep track of 3 categories in the SIR model

- **S**: susceptibles - who can get the disease
- **I**: infected - who have developed the disease and infect susceptibles
- **R**: recovered - who have recovered and become immune

Mathematical quantities: $S(t)$, $I(t)$, $R(t)$: no of people in each category

Goal: Find and solve equations for $S(t)$, $I(t)$, $R(t)$



0.5 The traditional modeling approach is very mathematical - our idea is to model, program and experiment

- Numerous books on mathematical biology treat the SIR model
- Quick modeling step (max 2 pages)
- Nonlinear differential equation model
- Cannot solve the equations, so focus is on discussing stability (eigenvalues), qualitative properties, etc.
- Very few extensions of the model to real-life situations

0.6 Dynamics in a time interval Δt : $\Delta t \beta SI$ people move from S to I

S-I interaction:

- In a mix of S and I people, there are SI possible pairs
- A certain fraction $\Delta t \beta$ of SI meet in a (small) time interval Δt , with the result that the infected “successfully” infects the susceptible
- The loss $\Delta t \beta SI$ in the S category is a corresponding gain in the I category

Remark. It is reasonable that the fraction depends on Δt (twice as many infected in $2\Delta t$ as in Δt). β is some unknown parameter we must measure, supposed to not depend on Δt , but maybe time t . β lumps a lot of biological and sociological effects into one number.

0.7 For practical calculations, we must express the S-I interaction with symbols

Loss in $S(t)$ from time t to $t + \Delta t$:

$$S(t + \Delta t) = S(t) - \Delta t \beta S(t)I(t)$$

Gain in $I(t)$:

$$I(t + \Delta t) = I(t) + \Delta t \beta S(t)I(t)$$

0.8 Modeling the interaction between R and I

R-I interaction:

- After some days, the infected has recovered and moves to the R category
- A simple model: in a small time Δt (say 1 day), a fraction $\Delta t \nu$ of the infected are removed (ν must be measured)

We must subtract this fraction in the balance equation for I :

$$I(t + \Delta t) = I(t) + \Delta t \beta S(t)I(t) - \Delta t \nu I(t)$$

The loss $\Delta t \nu I$ is a gain in R :

$$R(t + \Delta t) = R(t) + \Delta t \nu I(t)$$

0.9 We have three equations for S , I , and R

$$S(t + \Delta t) = S(t) - \Delta t \beta S(t)I(t) \quad (1)$$

$$I(t + \Delta t) = I(t) + \Delta t \beta S(t)I(t) - \Delta t \nu I(t) \quad (2)$$

$$R(t + \Delta t) = R(t) + \Delta t \nu I(t) \quad (3)$$



Before we can compute with these, we must

- know β and ν
- know $S(0)$ (many), $I(0)$ (few), $R(0)$ (0?)
- choose Δt

0.10 The computation involves just simple arithmetics

- Set $\Delta t = 6$ minutes
- Set $\beta = 0.0013$, $\nu = 0.008333$
- Set $S(0) = 50$, $I(0) = 1$, $R(0) = 0$

$$S(\Delta t) = S(0) - \Delta t \beta S(0)I(0) \approx 49.99$$

$$I(\Delta t) = I(0) + \Delta t \beta S(0)I(0) - \Delta t \nu I(0) \approx 1.002$$

$$R(\Delta t) = R(0) + \Delta t \nu I(0) \approx 0.0008333$$

- We can continue, but quickly gets boring...
- Solve with a for-loop as usual

0.11 First, some handy notation

$$S^n = S(n\Delta t), \quad I^n = I(n\Delta t), \quad R^n = R(n\Delta t)$$

$$S^{n+1} = S((n+1)\Delta t), \quad I^{n+1} = I((n+1)\Delta t), \quad R^{n+1} = R((n+1)\Delta t)$$

The equations can now be written more compactly (and computer friendly):

$$S^{n+1} = S^n - \Delta t \beta S^n I^n \tag{4}$$

$$I^{n+1} = I^n + \Delta t \beta S^n I^n - \Delta t \nu I^n \tag{5}$$

$$R^{n+1} = R^n + \Delta t \nu I^n \tag{6}$$

0.12 With variables, arrays, and a loop we can program

Suppose we want to compute until $t = N\Delta t$, i.e., for $n = 0, 1, \dots, N-1$. We can store $S^0, S^1, S^2, \dots, S^N$ in an array (or list).

```
from numpy import linspace, zeros
t = linspace(0, N*dt, N+1) # all time points
S = zeros(N+1)
I = zeros(N+1)
R = zeros(N+1)

for n in range(N):
    S[n+1] = S[n] - dt*beta*S[n]*I[n]
    I[n+1] = I[n] + dt*beta*S[n]*I[n] - dt*nu*I[n]
    R[n+1] = R[n] + dt*nu*I[n]
```

0.13 Here is the complete program

```
beta = 0.0013
nu = 0.008333
dt = 0.1 # 6 min (time measured in hours)
D = 30 # simulate for D days
N = int(D*24/dt) # corresponding no of hours

from numpy import zeros, linspace
t = linspace(0, N*dt, N+1)
S = zeros(N+1)
I = zeros(N+1)
R = zeros(N+1)
S[0] = 50
I[0] = 1

for n in range(N):
    S[n+1] = S[n] - dt*beta*S[n]*I[n]
```

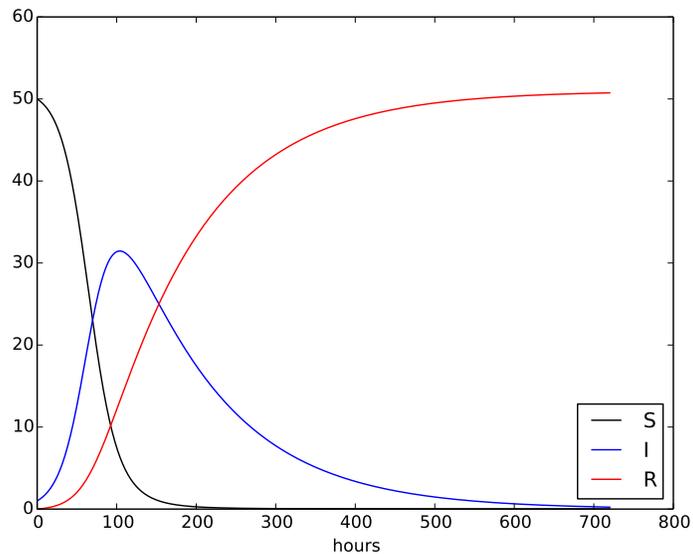
```

I[n+1] = I[n] + dt*beta*S[n]*I[n] - dt*nu*I[n]
R[n+1] = R[n] + dt*nu*I[n]

# Plot the graphs
from matplotlib.pyplot import *
plot(t, S, 'k-', t, I, 'b-', t, R, 'r-')
legend(['S', 'I', 'R'], loc='lower right')
xlabel('hours')
show()

```

0.14 We have predicted a disease!



0.15 The standard mathematical approach: ODEs

We had from intuition established

$$\begin{aligned}
S(t + \Delta t) &= S(t) - \Delta t \beta S(t) I(t) \\
I(t + \Delta t) &= I(t) + \Delta t \beta S(t) I(t) - \Delta t \nu I(t) \\
R(t + \Delta t) &= R(t) + \Delta t \nu I(t)
\end{aligned}$$

The mathematician will now make *differential equations*. Divide by Δt and rearrange:

$$\begin{aligned}\frac{S(t + \Delta t) - S(t)}{\Delta t} &= -\beta S(t)I(t) \\ \frac{I(t + \Delta t) - I(t)}{\Delta t} &= \beta t S(t)I(t) - \nu I(t) \\ \frac{R(t + \Delta t) - R(t)}{\Delta t} &= \nu R(t)\end{aligned}$$

0.16 A derivative arises as $\Delta t \rightarrow 0$

In any calculus book, the derivative of S at t is defined as

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

If we let $\Delta t \rightarrow 0$, we get derivatives on the left-hand side:

$$\begin{aligned}S'(t) &= -\beta S(t)I(t) \\ I'(t) &= \beta t S(t)I(t) - \nu I(t) \\ R'(t) &= \nu R(t)\end{aligned}$$

This is a 3x3 system of differential equations for the functions $S(t)$, $I(t)$, $R(t)$. For a unique solution, we need $S(0)$, $I(0)$, $R(0)$.

0.17 ODE system cannot be solved analytically

Recall the Forward Euler method: Approximate the derivative with a *finite difference*, e.g.,

$$S'(t) \approx \frac{S(t + \Delta t) - S(t)}{\Delta t}$$

and rearrange to get formulas like

$$S(t + \Delta t) = S(t) - \Delta t \beta S(t)I(t).$$

This brings us back to the first model, which we solved using a `for`-loop.

0.18 Parameter estimation is needed for predictive modeling

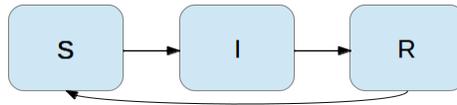
- Any small Δt will do
- One can reason about ν and say that $1/\nu$ is the mean recovery time for the disease (e.g., 1 week for a flu)
- β must in some way be measured, but we don't know what it means...

So, what if we don't know β ?

- Can still learn about the *dynamics* of diseases
- Can find the sensitivity to and influence of β
- Can apply *parameter estimation* procedures to fit β to data

0.19 Let us extend the model: no life-long immunity

Assumption. After some time, people in the R category lose the immunity. In a small time Δt this gives a leakage $\Delta t \gamma R$ to the S category. ($1/\gamma$ is the mean time for immunity.)



$$S^{n+1} = S^n - \Delta t \beta S^n I^n + \Delta t \gamma R^n \quad (7)$$

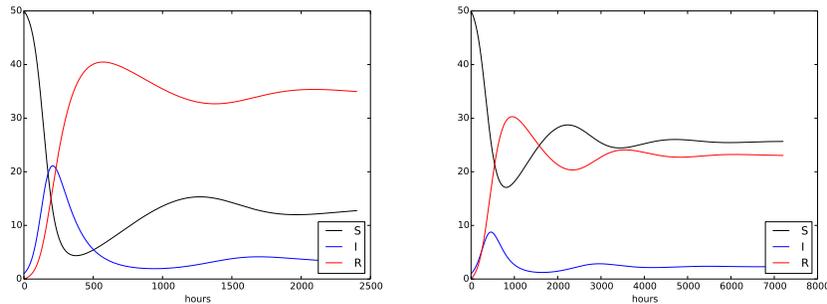
$$I^{n+1} = I^n + \Delta t \beta S^n I^n - \Delta t \nu I^n \quad (8)$$

$$R^{n+1} = R^n + \Delta t \nu I^n - \Delta t \gamma R^n \quad (9)$$

No complications in the computational model!

0.20 The effect of loss of immunity

$1/\gamma = 50$ days. β reduced by 2 and 4 (left and right, resp.):



0.21 And now for something similar: zombification!



Zombification: The disease that turns you into a zombie.

0.22 Zombie modeling is almost the same as SIR modeling

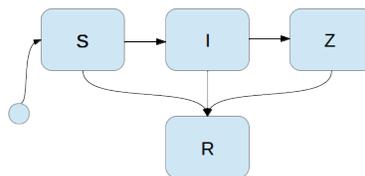
Categories.

1. S: susceptible humans who can become zombies
2. I: infected humans, being bitten by zombies
3. Z: zombies
4. R: removed individuals, either conquered zombies or dead humans

Mathematical quantities: $S(t)$, $I(t)$, $Z(t)$, $R(t)$

Zombie movie: *The Night of the Living Dead*, George A. Romero, 1968

0.23 Dynamics of the zombie SIZR model



1. Susceptibles are infected by zombies: $-\Delta t \beta S Z$ in time Δt (cf. the $\Delta t \beta S I$ term in the SIR model).

2. Susceptibles die naturally or get killed and then enter the removed category. The no of deaths in time Δt is $\Delta t \delta_S S$.
3. We also allow new humans to enter the area with zombies (necessary in a war on zombies): $\Delta t \Sigma$ during a time Δt .
4. Some infected turn into zombies (Z): $\Delta t \rho I$, while others die (R): $\delta_I \Delta t I$.
5. Nobody from R can turn into Z (important - otherwise zombies win).
6. Killed zombies go to R: $\Delta t \alpha S Z$.

0.24 The four equations in the SIZR model for zombification

$$\begin{aligned}
 S^{n+1} &= S^n + \Delta t \Sigma - \Delta t \beta S^n Z - \Delta t \delta_S S^n \\
 I^{n+1} &= I^n + \Delta t \beta S^n Z^n - \Delta t \rho I^n - \Delta t \delta_I I^n \\
 Z^{n+1} &= Z^n + \Delta t \rho I^n - \Delta t \alpha S^n Z^n \\
 R^{n+1} &= R^n + \Delta t \delta_S S^n + \Delta t \delta_I I^n + \Delta t \alpha S^n Z^n
 \end{aligned}$$

Interpretation of parameters:

- Σ : no of new humans brought into the zombified area per unit time.
- β : the probability that a theoretically possible human-zombie pair actually meets physically, during a unit time interval, with the result that the human is infected.
- δ_S : the probability that a susceptible human is killed or dies, in a unit time interval.
- δ_I : the probability that an infected human is killed or dies, in a unit time interval.
- ρ : the probability that an infected human is turned into a zombie, during a unit time interval.
- α : the probability that, during a unit time interval, a theoretically possible human-zombie pair fights and the human kills the zombie.

0.25 Simulate a zombie movie!

Three fundamental phases.

1. The initial phase (4 h)
2. The hysteric phase (24 h)
3. The counter attack phase (5 h)



How do we do this? As p in the vaccination campaign - the parameters take on different constant values in different time intervals.

H. P. Langtangen, K.-A. Mardal and P. Røtnes: Escaping the Zombie Threat by Mathematics, in A. Whelan et al.: *Zombies in the Academy - Living Death in Higher Education*, University of Chicago Press, 2013

0.26 Effective war on zombies

Introduce attacks on zombies at selected times T_0, T_1, \dots, T_m .

Model: Replace α by

$$\alpha_0 + \omega(t),$$

where α_0 is constant and $\omega(t)$ is a series of Gaussian functions (peaks) in time:

$$\omega(t) = a \sum_{i=0}^m \exp\left(-\frac{1}{2} \left(\frac{t - T_i}{\sigma}\right)^2\right)$$

Must experiment with values of a (strength), σ (duration is 6σ), point of attacks (T_i) - with proper values humans beat the zombies!

0.27 Summary

- A complex spreading of diseases can be modeled by intuitive, simple accounting of movement between categories

- Such models are known as *compartment models*
- Result: difference equations that are easy to simulate on a computer
- (Can let $\Delta t \rightarrow 0$ and get differential equations)
- Easy to add new effects (vaccination, campaigns, zombification)

All these slides and associated programs are available. Site: <https://github.com/hplgit/disease-modeling>. Just do

```
git clone https://github.com/hplgit/disease-modeling.git
```