

ABOUT PARAMETERS ON THE SIRD MODEL

MIGUEL MORENO

1. INTRODUCTION

In this notebook we use Python to simulate a model of epidemiology, the model that we will use is a simplification of the SIRD model with vaccine. The goal of this notebook is to show how small changes in the parameters may produce big changes in the simulation.

This notebook is to show an example of how mathematics is applied to other fields. It is not a notebook about epidemiology.

2. ACKNOWLEDGMENT

This notebook is based on the implementation of the SIR model made by Francisco Camacho in Platzi.

3. THE MODEL

We will work with a SIRD model with vaccination. We will use the following functions in our model:

- $S(t)$ the susceptible people at the time t ;
- $I(t)$ the infected people at the time t ;
- $R(t)$ the recovered people at the time t ;
- $D(t)$ the dead people at the time t ;
- $V(t)$ the vaccinated people at the time t .

The model will have the following parameters:

- N the size of the population;
- β the infection rate;
- μ the recovery rate;
- δ the mortality rate;
- ν the vaccinating rate.

The model is described by the following equations:

- The variation of $S(t)$ is given by the contact between infected and susceptible people, and the vaccinated people:

$$\frac{dS}{dt} = -\beta \frac{S(t)I(t)}{N} - \nu S(t).$$

- The variation of $I(t)$ is given by the contact between infected and susceptible people, the recovered people, and the death people:

$$\frac{dI}{dt} = \beta \frac{S(t)I(t)}{N} - \mu I(t) - \delta I(t).$$

Date: April 18, 2021.

- The variation of $R(t)$ is given by μ :

$$\frac{dR}{dt} = \mu I(t).$$

- The variation of $D(t)$ is given by δ :

$$\frac{dD}{dt} = \delta I(t).$$

- The variation of $V(t)$ is given by ν :

$$\frac{dV}{dt} = \nu S(t).$$

We will study the variation of the parameters of the models. The aim with this study is to understand that the information such as mortality rate of an infectious illness may not be enough to understand the possible damage that the illness may cause.

4. A NOT DEADLY VIRUS

Let us start by studying the parameters of a not deadly virus. In this case we can assume $\delta = \nu = 0$. With these assumptions we can only focus on two parameters β and μ and the relation between them.

In this case, the variation of infected people is given by

$$\frac{dI}{dt} = \beta \frac{S(t)I(t)}{N} - \mu I(t).$$

Therefore, whether $I(t)$ decreases or increases depends on the whether $\frac{dI}{dt}$ is positive, negative or zero. So the infected people will start reducing when

$$\beta \frac{S(t)I(t)}{N} - \mu I(t) < 0$$

since $I(0) > 0$ (otherwise the virus cannot infect new people), this is equivalent to

$$(1) \quad \beta \frac{S(t)}{N} < \mu.$$

Therefore $I(t)$ reach the biggest value when $S(t) = \frac{\mu N}{\beta}$. Let us take a close look to this with a population of $N = 1000$ and an initial infected individual $I(0) = 1$.

In Figure 1 (A) we can see the simulation of the model for the case when $\beta = \mu$, for graphical reasons we set $\beta = 0.3$.

In this case, when $t = 2$ the inequality (1) holds. We can see in Figure 1 (A) that the number of infected people stays very close to one. Therefore, there are very few infected people at the same time after $t = 2$. This does not mean that new infections are not appearing, what it is telling us is that people are getting infected and recovery at a very similar rate as we can see with the blue and green line. It is natural to ask whether the number of infected people will drop to zero before the susceptible number reach zero. To answer this question, we will increase the time of the simulation (see Figure 1 (B)).

As we can see in Figure 1 (B), the number of recovered people became constant with the time before the susceptible amount reach zero. It is clear that the bigger $\frac{\mu}{\beta}$ is, the faster the recovered people became constant.

In Figure 1 (C) we can see the simulation the case when $\frac{\mu}{\beta} < 1$. For graphical reasons we set $\beta = 0.3$ and $\mu = 0.05$.

It is clear from Figure 1 (C) that the smaller $\frac{\mu}{\beta}$ is, then is the bigger the maximum value of $I(t)$.

Now, by reducing β we simulate the reduction on the contacts between infected and susceptible individuals. In Figure 1 (D) we can see the simulation when β is reduced to $\beta = 0.2$.

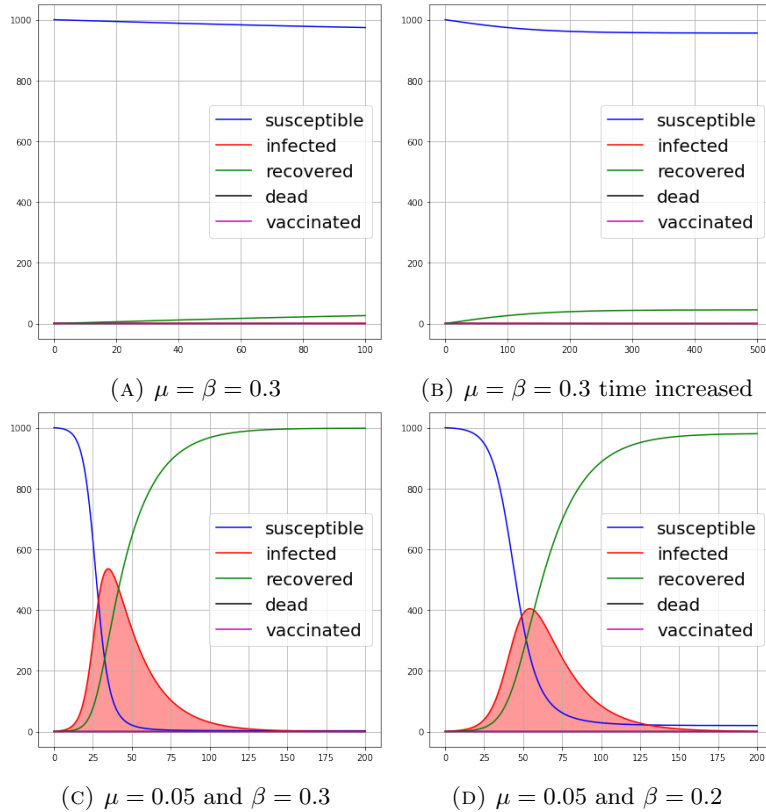


FIGURE 1. Simulation of a not deadly virus

As we have seen in this section, when the virus is not deadly, the value of $\frac{\mu}{\beta}$ is the information that determines the highest amount of infected individuals and when the infection of new individuals stops.

From now on we will assume that the values of β and μ are given by 0.3 and 0.05 respectively, and β can be reduced to 0.2 to simulate a reduction in the contacts between infected and susceptibles individuals.

5. A DEADLY VIRUS WITHOUT VACCINE

In this section we will study the parameters of a deadly virus, i.e. $\delta \neq 0$. We will assume there is no vaccine for the virus, see next section for $\nu \neq 0$. In this section we will focus on the variation of δ , at first sight we could think that the higher is δ , the more dangerous is the virus, i.e. the the amount of dead individuals will be higher at the end of the simulation.

The first thing we noticed is that the inequality (1) does not apply for the purposes of this section. Let us determine the equivalent of this inequality for this section.

In this case, the variation of infected people is given by

$$\frac{dI}{dt} = \beta \frac{S(t)I(t)}{N} - \mu I(t) - \delta I(t).$$

Therefore, whether $I(t)$ decreases or increases depends on the whether $\frac{dI}{dt}$ is positive, negative or zero. So the infected people will start reducing when

$$\beta \frac{S(t)I(t)}{N} - \mu I(t) - \delta I(t) < 0$$

since $I(t) > 0$, this is equivalent to

$$(2) \quad \beta \frac{S(t)}{N} < \mu + \delta.$$

Therefore $I(t)$ reach the biggest value when $S(t) = \frac{N(\mu+\delta)}{\beta}$. Let us take a close look to this with a population of $N = 1000$ and an initial infected individual $I(0) = 1$.

In Figure 2 (A) we can see the simulation of the model for a mortality rate of 0.075.

From Figure 2 (A) we can see that at the end of the simulation the amount of dead individuals is over 500. At first sight we can think that if we double δ , then the amount of dead individuals would increase. On the other hand, from the inequality (2) we could conclude that if δ is double, then the amount of infected individuals starts reducing at a smaller t . This observation can be explain by the fact that at a higher δ infected individuals have less time to infect others. On the other hand a smaller δ implies infected individuals have more time to infect other, but at the same time more time to increase the possibility to recover.

As we can see, it is not clear how the amount of dead individuals changes depending on the variation of δ . To get a better understanding of this relation, let us run a simulation with $\delta = 0.04$ and $\delta = 0.15$ (see Figure 2 (B) and (C), respectively).

As we can see in Figure 2 (B) and (C), in both cases the amount of dead individuals at the end of the simulation is smaller than when $\delta = 0.75$. In both cases the amount of dead individuals at the end of the simulation is very similar. One of the differences that we can see from Figure 2 (B) and (C) is that when $\delta = 0.04$ more people get recovered than when $\delta = 0.15$. This means that more individuals got infected when $\delta = 0.04$ and at the same time a similar amount of deaths. We conclude that it cannot be determined which case is more dangerous.

Let us assume that the value of δ is 0.075. In Figure 2 (D) we can see the simulation when the contacts are reduced, i.e. $\beta = 0.2$.

As we have seen in this section, when the virus is deadly without a vaccine, the value of $\frac{\mu+\delta}{\beta}$ is the information that determines the highest amount of infected individuals and when the infection of new individuals stops. But not much can be deduced from the variation of δ .

From now on we will assume that the values of δ is 0.075.

6. A DEADLY VIRUS WITH VACCINATION

In this section we will study the last parameter of our model, i.e. the vaccination rate $\nu \neq 0$. Even though a vaccination campaign is more complicated than only one

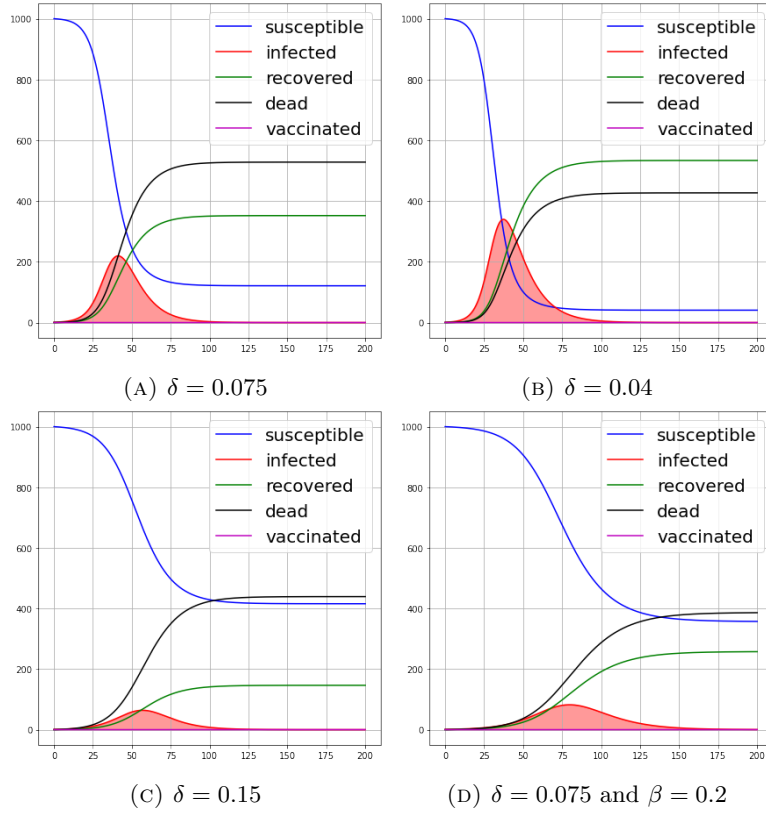


FIGURE 2. Simulation of a deadly virus without vaccine

parameter (effectiveness of the vaccine, speed of the vaccination campaign, etc), we will run a simplification of this by only having one parameter.

The first thing we noticed is that the inequality (2) applies for the purposes of this section. This is because in our model the variation of the infected individuals is not explicitly related to the vaccination rate, the relation between the infected individuals and the vaccination rate comes from the variation of the susceptible individuals. Therefore we will focus on the variation of the susceptible individuals.

In this case, the variation of susceptible people is given by

$$(3) \quad \frac{dS}{dt} = -\beta \frac{S(t)I(t)}{N} - \nu S(t)$$

It is clear from (3) that $\frac{dS}{dt}$ is always negative unless $S(t)$ is zero, this can be easily seen from the fact that

$$-\beta \frac{S(t)I(t)}{N}$$

is never positive and only zero when $S(t)$ or $I(t)$ is zero and

$$-\nu S(t)$$

is never positive and only zero when $S(t)$. So $S(t)$ will decrease until it reach zero, it will decrease faster than in the previous case and as a consequence the maximum of $I(t)$ will be smaller than in the previous case.

Let us recall the graphic when $\beta = 0.3$, $\mu = 0.05$, $\delta = 0.075$, and $\nu = 0$ (see Figure 3 (A)). In Figure 3 (B) we can see the simulation when we set $\nu = 0.01$.

As we can see, the reduction on the amount of dead individuals and the infected population is notorious when $\nu = 0.01$. Even though ν is does not appear as a parameter in (2), a small variation on ν implies a notorious change on the maximum of $I(t)$ and more important on the maximum of $D(t)$. Let us finish by the simulation when the contacts are reduced, i.e. $\beta = 0.2$ and the vaccination rate is 0.01 (see Figure 3 (C)).

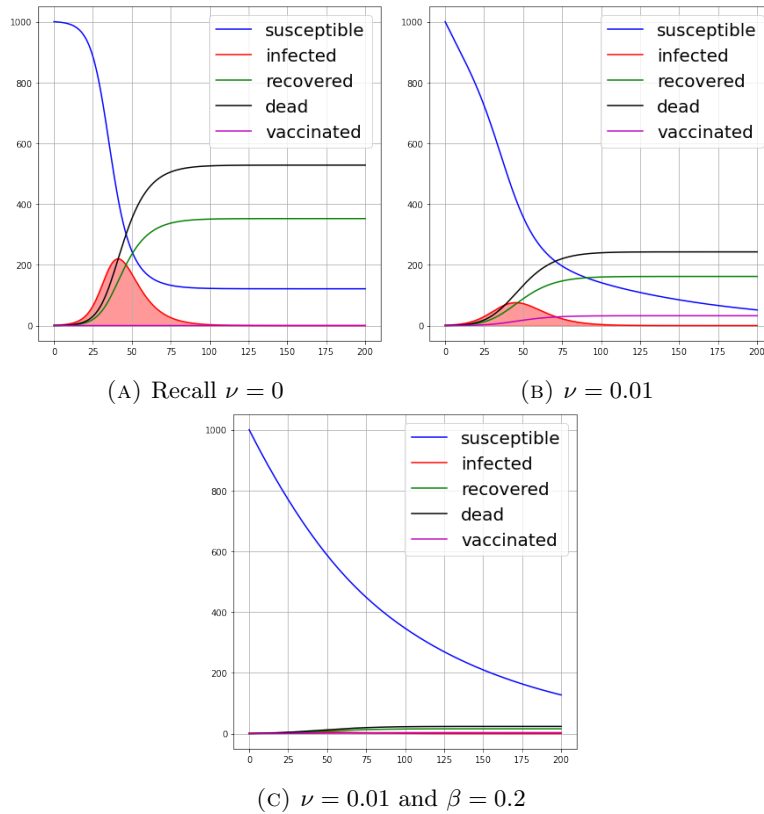


FIGURE 3. Simulation of a deadly virus with vaccination

After running simulations with different values on the parameters, we have seen that small changes on the parameters of the model may imply drastically changes on the simulation. Therefore, it is not possible to make trustable conclusion based on the variation of one parameter. In this notebook we have seen the importance of having the right values of the parameters and trustable data.

Notice that this is a simple model in which many other parameters have been excluded, for example the natality rate, a segmentation of the population by age, etc.

7. APPENDIX: THE CODE

The implementation of the model for the simulation was the following.

```

import numpy as np
import matplotlib.pyplot as plt

def rk4vec(t0, y0, dt, f):
    k1 = f(t0, y0)
    k2 = f(t0 + dt/2.0, y0 + dt * k1 / 2.0)
    k3 = f(t0 + dt/2.0, y0 + dt * k2 / 2.0)
    k4 = f(t0 + dt, y0 + dt * k3)
    y = y0 + (dt/6.0)*(k1 + 2.0*k2 + 2.0*k3 + k4)
    return y

class SIRDV_model():

    def __init__(self, init_condition, tmin = 0., tmax = 50., n = 10000, **params):

        self.tmin = tmin
        self.tmax = tmax
        self.n = n
        self.t = np.linspace(self.tmin, self.tmax, self.n)
        self.dt = self.t[1] - self.t[0]
        self.S = np.zeros([self.n])
        self.I = np.zeros([self.n])
        self.R = np.zeros([self.n])
        self.D = np.zeros([self.n])
        self.V = np.zeros([self.n])
        self.set_params(**params)
        self.init_condition = init_condition

    def set_params(self, beta, mu, delta, nu, max_population):

        self.beta = beta
        self.mu = mu
        self.delta = delta
        self.nu = nu
        self.max_population = max_population

    def func(self, t, u):

        self.uprime = np.zeros_like(u)
        self.uprime[0] = -self.beta*u[0]*u[1]/max_population-self.nu*u[0]
        self.uprime[1] = self.beta*u[0]*u[1]/max_population-self.mu*u[1]-self.delta*u[1]
        self.uprime[2] = self.mu*u[1]
        self.uprime[3] = self.delta*u[1]
        self.uprime[4] = self.nu*u[1]
        return self.uprime

    def run_solver(self):

        self.u0 = np.array(self.init_condition)
        self.u1 = np.zeros_like(self.u0)
        for i in range(self.n):

```

```
self.S[i] = self.u0[0]
self.I[i] = self.u0[1]
self.R[i] = self.u0[2]
self.D[i] = self.u0[3]
self.V[i] = self.u0[4]
self.u1 = rk4vec(self.t[i], self.u0, self.dt, self.func)
self.u0 = np.copy(self.u1)
```

Github: [MIGUELWAN](#)

E-mail: CONTACT@MIGUELMATH.COM

Webpage: [HTTP://MIGUELMATH.COM](http://MIGUELMATH.COM)