Modelling the effect of vaccinations using differential equations

Vince Knight

1 Introduction

The World Health Organisation estimates that the measles vaccination has saved more than 17 million lives since 2000 [1]. It is possible to model the effect of vaccination using differential equations. The model considered here is called an SIR model which is a compartementalised model of infection where individuals can be in one of 3 states:

- Susceptible (S): members of the population who may become infected;
- Infected (I): infected members of the population who will eventually recover;
- Recovered (R): recovered members of the population, not that mathematically death is here equivalent to recovery.

Figure 1 shows this diagrammatically.

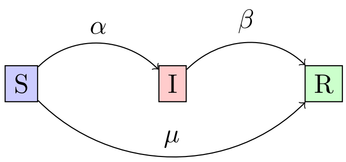

Figure 1: The SIR model

The parameters of the model shown in Figure 1 are:

- α the infection rate;
- β the recovery rate;
- μ the vaccination percentage;

This can be expressed mathematically:

$$\frac{dS}{dt} = -\alpha IS - \mu S \tag{1}$$

$$\frac{dI}{dt} = \alpha IS - \beta I \tag{2}$$

$$\frac{dR}{dt} = \mu S + \beta I \tag{3}$$

In the next section we will use Sympy to attempt to solve these equations analytically:

2 (Not) finding an exact solution

It is possible to use Sympy to solve systems of differential equations, however when attempting to do this here it appears to fail:

```
>>> import sympy as sym
>>> S, I, R = sym.Function("S"), sym.Function("I"), sym.Function("V")
>>> N, mu, alpha, beta, t = sym.symbols("N, mu, alpha, beta, t")
>>> eq1 = sym.Derivative(S(t), t) - (- alpha * S(t) * I(t) - mu * R(t))
>>> eq2 = sym.Derivative(I(t), t) - (alpha * I(t) * S(t) / N  - beta * I(t))
```

```
>>> eq3 = sym.Derivative(R(t), t) - (beta * I(t) + mu * R(t))
>>> sym.dsolve((eq1, eq2, eq3))
NotImplementedError                        Traceback (most recent call last)
...
```

I believe that this is due to the complexity of the differential equations that Sympy is unable to handle analytically. An exact solution of the model without vaccination is obtained in [2]. In the next section we will however solve these equations numerically.

3 Solving the equations numerically and the effect of vaccination rates

We can use a numerical integration technique to solve these equations numerically. The technical publication describing the specific algorithm used is described in [3].

First we create a function that gives expressions for the derivatives at any given point in time:

```
>>> def dx(x, t, alpha, beta, mu):
...     return (- alpha * x[1] * x[0] - mu * x[0],
...             alpha * x[1] * x[0]  - beta * x[1],
...             beta * x[1] + mu * x[0])
```

We can then plot a number of different scenarios as shown in Figure 2. Here is the code that corresponds to the first plot:

```
>>> alpha = 1 / 1000  # Every 1000 interactions leads to infection
>>> beta = 1 / 5  # take 5 time units to recover
>>> N = 10 ** 4  # Population of 10 thousand people
>>> mu = 0  # 0 vaccination rate
>>> ts = np.linspace(0, 10, 5000)
>>> xs = integrate.odeint(func=dx, y0=np.array([N - 1, 1, 0]), t=ts, args=(alpha, beta, mu))
>>> S, I, R = xs.T
>>> plt.figure()
>>> plt.plot(ts, S, label="Susceptibles")
>>> plt.plot(ts, I, label="Infected")
>>> plt.plot(ts, R, label="Recovered")
>>> plt.legend()
>>> plt.title(f"$\max(I)={round(max(I))}$ ($\\alpha={alpha}$, $\\beta={beta}$,
$\mu={mu}$)");
```

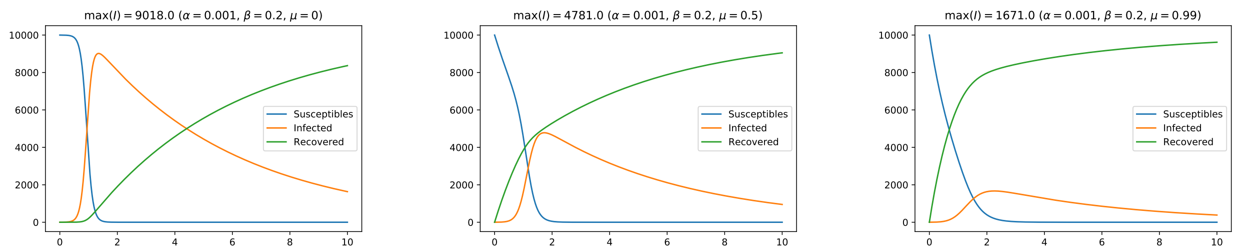

Figure 2: The evolution of the population for different vaccination rates

It is also possible to compute the maximum percentage of infection as a function of the vaccination rate:

```
>>> vaccination_rates = np.linspace(0, 1, 500)
>>> max_percent_of_infected = []
>>> for mu in vaccination_rates:
...     xs = integrate.odeint(func=dx, y0=np.array([N - 1, 1, 0]), t=ts, args=(alpha, beta, mu))
>>> S, I, R = xs.T
>>> max_percent_of_infected.append(max(I) / N)
>>> plt.figure()
>>> plt.plot(vaccination_rates, max_percent_of_infected)
>>> plt.xlabel("Vaccination rate")
>>> plt.ylabel("% of population infected");
```

This is shown in Figure 3.

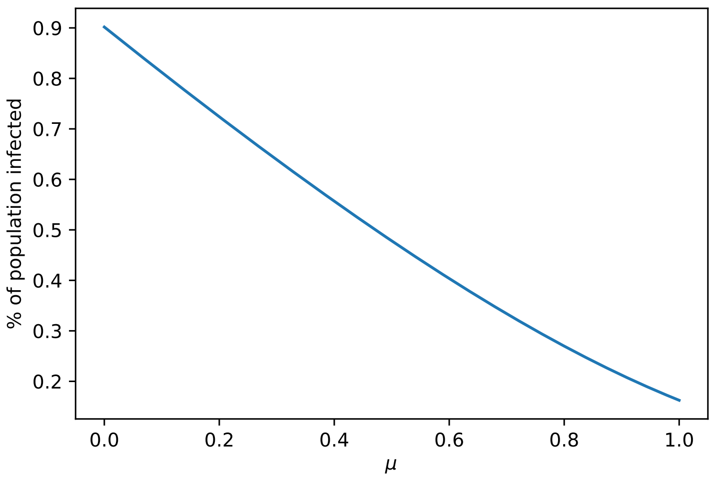

Figure 3: The effect of vaccination rate on maximum infection percentage

4 Conclusion

We see in our model that a large vaccination rate is required to ensure a high level of immunity (ie a low maximum level of total infection). This type of approach uses differential equations to model the interactions of individuals and the spread of disease, finally we solve these equations using techniques from numerical analysis.

References

[1] Measles vaccination has saved an estimated 17.1 million lives since 2000. `http://www.who.int/news-room/detail/12-11-2015-measles-vaccination-has-saved-an-estimated-17-1-million-lives-since-2000`. Accessed: 2018-09-04.

[2] Tiberiu Harko, Francisco SN Lobo, and MK Mak. Exact analytical solutions of the susceptible-infected-recovered (sir) epidemic model and of the sir model with equal death and birth rates. *Applied Mathematics and Computation*, 236:184–194, 2014.

[3] Krishnan Radhakrishnan and Alan C Hindmarsh. Description and use of lsode, the livermore solver for ordinary differential equations. 1993.