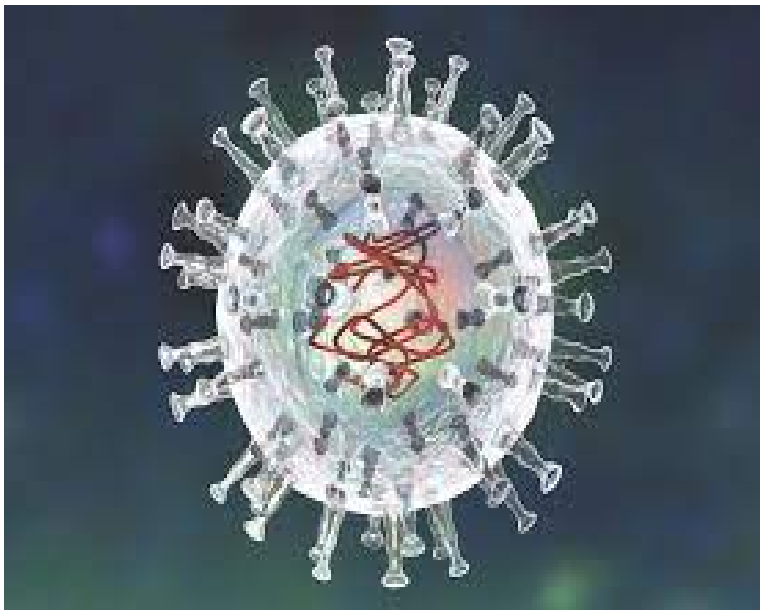
UNDERSTANDING EPIDEMICS
(MODELING WITH DIFFERENTIAL EQUATIONS)



Goal: A model to describe the spread of a disease in a population.

Getting Started :

- Open a new Jupyter notebook and save it as name_epidemicmodel.ipynb.
- **Useful Python commands:** `sympy.dsolve`, `sympy.lambdify`, `plot.subplots`, `for loop`, `numpy.arange`, `numpy.linspace`, `len`, `plot.scatter,...`, `solve_ivp`, `plot.quiver`, `numpy.meshgrid`

1. Conceptual Overview

Over the centuries, there have been several examples of how epidemics have significantly effect at humans. Some of the recent epidemics were AIDS, Covid-19, and Ebola. If we can understand the nature of how a disease spreads through a population, then we can be better equipped to contain it through vaccination or quarantine strategies. Alternatively, in the case of the biological control of pests, we may wish to determine how to increase the spread of the disease (for example, myxomatosis or calicivirus in rabbits) so as to find an efficient way of reducing the population. Unfortunately, humans themselves have been subjected to this means of control. Many diseases are spread by infected individuals in the population coming into close contact with susceptible individuals. These include influenza, measles, chickenpox, glandular fever and Covid-19. On the other hand, malaria is transmitted through a host, a mosquito, which carries the disease from individual to individual. Certain diseases are more contagious than others. Measles and influenza are highly contagious, whereas glandular fever is much less so. Many diseases, such as mumps and measles, confer a

lifelong immunity; however, influenza and typhoid have short periods of immunity and can be contracted more than once.

Model assumptions. When considering a disease, the population can be divided into distinct groups: susceptibles $S(t)$, contagious infectives $I(t)$ and recovers $R(t)$, where t denotes time. The susceptibles are those liable to catch the disease, while the infectives are those infected with the disease who are capable of spreading it to susceptibles. There are also those who have recovered from the disease and are no longer susceptible, who form a further separate group.
Initially, we make some assumptions and then build the model based on them:

- We assume that the populations of susceptibles and contagious infectives are large so that random differences between individuals can be neglected.
- We ignore births and deaths in this model and assume the disease is spread by contact.
- We neglect the latent period for the disease, setting it equal to zero.
- We assume all those who recover from the disease are then immune (at least within the time period considered).
- We also assume that, at any time, the population is homogeneously mixed; that is, we assume that the contagious infectives and susceptibles are always randomly distributed over the area where the population lives.

Compartmental model. The only way the number of susceptibles can change is the loss of those who become infected, as there are no births, and none of those who become contagious infectives can become susceptibles again. The number of infectives changes due to the susceptibles becoming infected and decreases due to those infectives who die, become immune or are quarantined. The latter cannot become susceptibles again (from the assumptions made). This is illustrated in the below compartmental diagram.

The appropriate word equations are

$$\left\{\begin{array}{c} rate\ of \\ change\ in\ no. \\ susceptibles \end{array}\right\} = -\left\{\begin{array}{c} rate\ of \\ susceptibles \\ infected \end{array}\right\} \tag{1.1}$$

$$\left\{\begin{array}{c} rate\ of \\ change\ in\ no. \\ infectives \end{array}\right\} = \left\{\begin{array}{c} rate \\ susceptibles \\ infected \end{array}\right\} - \left\{\begin{array}{c} rate\ of \\ infectives \\ recovered \end{array}\right\} \tag{1.2}$$

$$\left\{\begin{array}{c} rate\ of \\ change\ in\ no. \\ of\ recovered \end{array}\right\} = \left\{\begin{array}{c} rate\ of \\ infectives \\ recovered \end{array}\right\} \tag{1.3}$$

We must account for those removed from the system, in this case, those who have recovered from the disease. More generally, the removed can also consist of fatalities due to the disease, those who become immune to the disease, and those infectives

who are quarantined. The number of infectives removed in the time interval should not depend in any way upon the number of susceptibles, but only on the number of infectives. We assume that the rate at which infectives recover is directly proportional to the number of infectives and write

$$\left\{ \begin{array}{c} rate\ of \\ infectives \\ recovered \end{array} \right\} = \gamma I(t). \tag{1.4}$$

where γ is a positive constant of proportionality, called the recovery rate, or more generally, the removal rate.

To model the total rate of susceptibles infected, first consider the susceptibles infected by a single infective. It is evident that the greater the number of susceptibles, the greater the increase in the number of infectives. Thus, the rate of susceptibles infected by a single infective will be an increasing function of the number of susceptibles. For simplicity, let us assume that this rate is directly proportional to the number of susceptibles. If we denote the number of susceptibles at time t by $S(t)$, then the rate at which susceptibles are infected is $\lambda(t)S(t)$. Note, it is not necessary to treat λ as a constant. The total rate that susceptibles are infected is

$$\left\{ \begin{array}{c} rate\ of \\ susceptibles \\ infected \end{array} \right\} = \lambda(t)S(t). \tag{1.5}$$

Later, we will find a suitable dependence for λ on the number of infectives. The term $\lambda(t)$ is called the force of infection. We can also interpret $\lambda(t)$ as the instantaneous probability per unit time of a single susceptible becoming infected, and thus, $\lambda(t) = \beta I(t)$. where β is called a transmission coefficient.

Using the word equations (1.1),(1.2),(1.3),(1.4) and (1.5), we can write,

$$\left\{ \begin{array}{c} rate\ of \\ change\ in\ no. \\ susceptibles \end{array} \right\} = -\beta S(t)I(t)$$

$$\left\{ \begin{array}{c} rate\ of \\ change\ in\ no. \\ infectives \end{array} \right\} = \beta S(t)I(t) - \gamma I(t)$$

$$\left\{ \begin{array}{c} rate\ of \\ change\ in\ no. \\ of\ recovered \end{array} \right\} = \gamma I(t)$$

Hence, we can also write the governing equations as

$$\frac{dS}{dt} = -\beta SI \tag{1.6}$$

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{1.7}$$

$$\frac{dR}{dt} = \gamma I \tag{1.8}$$

Use the above model to answer the following questions.

Question 1. *Draw the direction field and the solution trajectories of the given system (1.6),(1.7) on the phase plane with 44% recovery rate and transmission coefficient equal to 2.18×10^{-3} (you may either do it analytically or write a code in Python). For reference, the following pseudo code may be helpful.*

Algorithm
// Input: ODE system, Initial Values, time interval, step-size
// Output: Plot the solution trajectories and direction fields

Step1: Import the libraries
`use numpy, sympy, scipy.integrate, matplotlib.plot`

#Plot the solution trajectories

Step2: Define the given system in the form of lambdify generated function

Step3: Input the interval of integration

Step4: Create an array for various initial conditions

Step5: Define the discrete-time space
`use np.linspace, np.arange`

Step6: for i from 1 to length of array in Step4
solve _ ivp
plot the solution trajectories
end for

#Plot the direction field

Step7: Define a function containing the system of differential equations

Step8: Find the maximum value of the plotted trajectories in Step6 over the x-axis and y-axis

Step9: Create a discrete space for x and y upto the obtained maximum values in Step8

Step10: Create a meshgrid using the discrete spaces obtained in Step9
`use np.meshgrid`

Step11: Find the slopes of functions using defined function in Step7 over the values of meshgrid obtained in Step10

Step12: Normalize the obtained slope points

Step13: Plot the direction field as an arrows from the point in meshgrid to the obtained slope points
`use plot.quiver(starting position points, slope direction points)`

Note: You may refer to the python code available on the course website for the numerical solvers (Euler, rk2, rk4, and ode45).

Question 2. *Using Euler, RK-2, RK-4 and ode45 methods, solve the given system* (1.6),(1.7), *and* (1.8) *numerically with* 44% *recovery rate and transmission coefficient equal to* 2.18×10^{-3} *over the interval* $[0, 10.5]$ *with step size* 0.7 *using the initial conditions* $S(0) = 762$, $I(0) = 800$ *and* $R(0) = 0$.

Question 3. *Choose the suitable value of transmission coefficient for which the given system* (1.6),(1.7), *and* (1.8) *becomes a linear system and then solve the linear system analytically. Also, verify your answer using* `dsolve`.

Question 4. *Modify the program in question (3), to find the particular solution of the linear system with a* 44% *recovery rate. Use the initial conditions* $S(0) = 762$, $I(0) = 800$ *and* $R(0) = 0$. *Plot the particular solution. Repeat this exercise using a numerical solver of your choice and compare the results by overlaying the solution on the same graph.*

Question 5. *Compare the numerical solution of the given non-linear system with the analytical solution of the linearized system obtained in question (3) by plotting for the time interval [0, 10.5]. Comment on any disparity that you may observe.*