A REPORT ON

STOCHASTIC DISEASE MODELLING

BITS PILANI

In partial fulfillment of the MATH F424 Course

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

July, 2020

2. Table of Contents

1. Cover Page 1
2. Table of Contents 2
3. Introduction 4
4. The SIS Model 5
    4.1. Model description 5
    4.2. Deterministic SIS model 6
    4.3. Formulation of DTMC epidemic model : 7
        4.3.1 Algorithm for Simulating The SIS Model on MATLAB 9
        4.3.2. Curve fitting 10
    4.4. Results 12
5. The SIR Model 14
    5.1. The SIR Model Construction 14
    5.2. The SIR Deterministic Model 15
    5.3. The SIR Discrete Time Markov Chain (DTMC) Model: 17
    5.4. The SIR Continuous Time Markov Chain (CTMC) Model: 18
        5.4.1. The Branching Process Approximation 19
        5.4.2. Gillespie Algorithm 22
    5.5. SIR Stochastic Differential Equations 25
        5.5.1. Derivation Of SDE 25
        5.5.2. Euler-Maruyama method 28
    5.6. The SIR Model for a Metapopulation 30
        5.6.1 The Cross Coupled Model 31
        5.6.2. The Mobility Model 32
    5.7. Estimation Of Parameters 35
    5.8. COVID-19 in India - An Analysis 37
        5.8.1. Data 37
        5.8.2. Overall Analysis 38
        5.8.3. Phase-wise Analysis 40
            5.8.3.1. Phase 1 41
            5.8.3.2. Phase 2 43
            5.8.3.3. Phase 3 45
6. Conclusion 47
7. Appendix 48
    SIS codes 48
    SIR codes 50
8. References 64

3. Introduction

In today's day and age, disease modelling holds prime importance. Using disease epidemic modelling, we can predict the dynamics of a disease such as its reproduction rate, probability of outbreak, duration of the epidemic and many more things. The main goal of epidemic models is the ability to accurately predict spreading patterns of a given communicable disease affecting a specific population. These models allow decision makers to assess the various intervention strategies available to them and to plan accordingly. Several approaches have been developed to model disease outbreaks, namely, compartmental equations, stochastic equations, agent-based simulations, etc. Each approach suits a particular aspect of the outbreak being studied, built upon hypotheses compatible with historical records. They include biological content of the disease, mechanisms behind disease transmission, social interactions among the target population and its spatial structure.

In this project, we discuss two types of models, the SIS model and the SIR model in detail. Different models are used for different diseases depending on their characteristics. The deterministic and stochastic versions are discussed and they have been compared. Various approaches and methods are also explained in a structured manner.

4. The SIS Model

4.1. Model description

We begin our discussion using the susceptible-infectious-susceptible (SIS) epidemic model. The SIS model describes the circulation of a single communicable disease in a susceptible population.
The transmission of the disease occurs when infectious people transmit the disease to healthy susceptible individuals.
The infectious diseases are caused by pathogenic microorganisms such as viruses, fungi or bacterias.The infectious period extends throughout the whole course of the disease until the recovery of the patient, warranting a two-stage model: either infected or susceptible. After recovery patients do not get immunity against disease so they are prone to get disease again when coming in contact with an infectious person.
The SIS epidemic model has been applied to sexually transmitted diseases.

We will be discussing stochastic modelling for the discrete time Markov chain (DTMC) of the SIS model and then will simulate the spread of pneumococcus amongst children aged two years and under in scotland.
In a DTMC model, the time and the state variables are discrete.

Assumptions:

- Population is assumed to be homogeneously mixed with an equal rate of getting infectious.
- There is no vertical transmission of the disease (all individuals are born susceptible) and there are no disease-related deaths.

Notations :

- Infected individuals are denoted by I.
- Population size is denoted by N
- Susceptible individual is denoted by S.
- Infected people recover to susceptible states with rate γ .
- The adequate interaction between an infected person with a susceptible, one get a new infection with rate α.

Model Framework :
In general the spread of an infectious disease depends upon the population, the mode of transmission of the disease and features of the disease.

Depending upon the level of pathogen, the population, in general is classified as:

- ➢ Susceptible population (healthy population which is prone to infection), denoted by S.
- ➢ Infectious population, denoted by I.
- ➢ Removed or immune population, denoted by R.

4.2. Deterministic SIS model

If $S(t)$ denotes the number of susceptibles at time t and $I(t)$ denotes the number of infecteds at time t then the spread of the disease is described by the pair of differential equations

$$\frac{dS}{dt} = \mu N - \beta S(t)I(t) + \gamma I(t) - \mu S(t)$$

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

with appropriate starting values $S(0)$ and $I(0)$ with

$$S(0) + I(0) = N$$

In these equations μ is the rate at which a single individual dies and γ is the per capita rate at which a single individual recovers. Hence assuming that the infectious period follows an exponential distribution the average infectious period is $\frac{\beta}{\gamma}$ which is the rate at which a single infected individual makes contact with and infects each susceptible individual.

A key concept in mathematical epidemiology is the idea of the basic reproduction R_0 number . This is defined as the expected number of secondary cases produced by a single newly infected individual entering a disease-free population at equilibrium. We find that $R_0 = \frac{\beta N}{(\mu+\gamma)}$.

The differential equations are equivalent to the well-known logistic equation for population growth and has a solution given by

$$I(t) = \begin{cases} \left[\left(\frac{\beta}{\beta N - \mu - \gamma}\right)\left(1 - e^{-(\beta N - \mu - \gamma)t}\right) + \frac{e^{-(\beta N - \mu - \gamma)t}}{I(0)}\right]^{-1}; R_0 \neq 1 \\ \left[\beta t + \frac{1}{I(0)}\right]^{-1}; R_0 = 1 \end{cases}$$

Hence if $R_0 \leq 1$, $I(t) \to 0$ as $t \to \infty$ whereas $R_0 > 1$, $I(t) \to N(1 - \frac{1}{R_0})$ as $t \to \infty$.

4.3. Formulation of DTMC epidemic model :

Let $S(t),\ I(t)$ and $R(t)$ denote discrete random variables for the number of susceptible, infected, and immune individuals at time t, respectively. In a DTMC epidemic model, $t \in \{0, \Delta t, 2\Delta t, ...\}$ and the discrete random variables satisfy

$$S(t)\ , I(t),\ R(t)\ \in \{0, 1, 2, ..., N\}$$

In an SIS epidemic model, there is only one independent random variable, I(t), because $S(t) = N - I(t)$, where N is the constant total population size. The $\{I(t)\}_{t=0}^{\infty}$ stochastic process has an associated probability function,

$$p_i(t)\ =\ Prob\{I(t) = i\}$$

for $i \in \{0, 1, 2, ..., N\}$ and $t \in \{0, \Delta t, 2\Delta t, ...\}$,where

$$\sum_{i=0}^{N} p_i(t) = 1$$

Let $p(t) = (p_0(t), p_1(t), ..., p_N(t))^T$ denote the probability vector associated with $I(t)$. The stochastic process has the Markov property if

$$Prob\{I(t + \Delta t)|I(0), I(\Delta t), ..., I(t)\} = Prob\{I(t + \Delta t)|I(t)\}$$

The Markov property means that the process at time $t + \Delta t$ only depends on the process at the previous time step t.

To complete the formulation for a DTMC SIS epidemic model, the relationship between the states $I(t)$ and $I(t + \Delta t)$ needs to be defined. This relationship is determined by the underlying assumptions in the SIS epidemic model and is defined by the transition matrix. The probability of a transition from state $I(t) = i$ to state $I(t + \Delta t) = j$ i.e. $i \rightarrow j$ in time Δt is denoted as

$$p_{ji}(t + \Delta t, t) = Prob\{I(t + \Delta t) = j|I(t) = i\}$$

When the transition probability $p_{ji}(t + \Delta t, t)$ does not depend on t and $p_{ji}(\Delta t)$, the process is said to be time homogeneous. For the stochastic SIS epidemic model, the process is time homogeneous because the deterministic model is autonomous.

To reduce the number of transitions in time Δt, we make one more assumption. The time step Δt is chosen sufficiently small such that the number of infected individuals changes by at most one during the time interval Δt, that are

$$i \to i+1,\ i \to i-1 \text{ or } i \to i$$

Either there is a new infection, a birth, a death, or a recovery during the time interval Δt. Of course, this latter assumption can be modified, if the time step cannot be chosen arbitrarily small. In this latter case, transition probabilities need to be defined for all possible transitions that may occur, $i \to i+2,\ i \to i+3$, etc. In the simplest case, with only three transitions possible, the transition probabilities are defined using the rates (multiplied by Δt) in the deterministic SIS epidemic model. The transition probabilities for the DTMC epidemic model satisfy

$$p_{ji}(\Delta t) = \begin{cases} \dfrac{\beta i(N-i)}{N}\Delta t, & j = i+1 \\ (b+\gamma)i\Delta t, & j = i-1 \\ 1 - \left[\dfrac{\beta i(N-i)}{N} + (b+\gamma)i\right]\Delta t, & j = i \\ 0, & j \neq i+1, i, i-1 \end{cases}$$

Here β > 0 is the contact rate, γ > 0 is the recovery rate, b ≥ 0 is the birth rate.The probability of a new infection, $i \to i+1$, is $\frac{\beta i(N-i)\Delta t}{N}$. The probability of a death or recovery, $i \to i-1$, is $(b+\gamma)i\Delta t$. Finally, the probability of no change in state, $i \to i$, is $1 - [\frac{\beta i(N-i)}{N} + (b+\gamma)i]\Delta t$. Since a birth of a susceptible must be accompanied by a death, to keep the population size constant, this probability is not needed in either the deterministic or stochastic formulations.

The sum of the three transitions equals one because these transitions represent all possible changes in the state i during the time interval Δt. To ensure that these transition probabilities lie in the interval $[0,1]$, the time step Δt must be chosen sufficiently small such that

Applying the Markov property and the preceding transition probabilities, the probabilities $p_i(t+\Delta t)$ can be expressed in terms of the probabilities at time t. At time $t+\Delta t$,

$$p_i(t+\Delta t) = p_{i-1}(t)\beta(i-1)\Delta t + p_{i+1}(t)d(i+1)\Delta t + p_i(t)(1-[b(i)+d(i)]\Delta t)$$

for $i = 1, 2, \ldots N$ where $b(i) = \frac{\beta i(N-i)}{N}$ and $d(i) = (b+\gamma)i$

A transition matrix is formed when the states are ordered from 0 to N . For example, the $(1,1)$ element in the transition matrix is the transition probability from state zero to state zero, $p_{00}(\Delta t) = 1$, and the $(N+1, N+1)$ element is the transition probability from state N to state N , $p_{NN}(\Delta t) = 1 - (b+\gamma)N\Delta t = 1 - d(N)\Delta t$. Denote the transition matrix as $P(\Delta t)$. Matrix $P(\Delta t)$ is a $(N+1)\times(N+1)$ tridiagonal matrix given by

$$\begin{pmatrix} 1 & d(1)\Delta t & 0 & \cdots & 0 & 0 \\ 0 & 1-(b+d)(1)\Delta t & d(2)\Delta t & \cdots & 0 & 0 \\ 0 & b(1)\Delta t & 1-(b+d)(2)\Delta t & \cdots & 0 & 0 \\ 0 & 0 & b(2)\Delta t & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d(N-1)\Delta t & 0 \\ 0 & 0 & 0 & \cdots & 1-(b+d)(N-1)\Delta t & d(N)\Delta t \\ 0 & 0 & 0 & \cdots & b(N-1)\Delta t & 1-d(N)\Delta t \end{pmatrix}$$

The DTMC SIS epidemic process $\{I(t)\}_{t=0}^{\infty}$ is now completely formulated. Given an initial probability vector $p(0)$, it follows that $p(\Delta t) = P(\Delta t)p(0)$ .The probability can be expressed in matrix and vector notation is

$$p(t+\Delta t) = P(\Delta t)p(t) = P_{n+1}(\Delta t)p(0) \text{ where } t = n\Delta t .$$

4.3.1 Algorithm for Simulating The SIS Model on MATLAB

We sample 3 paths of the DTMC SIS process.

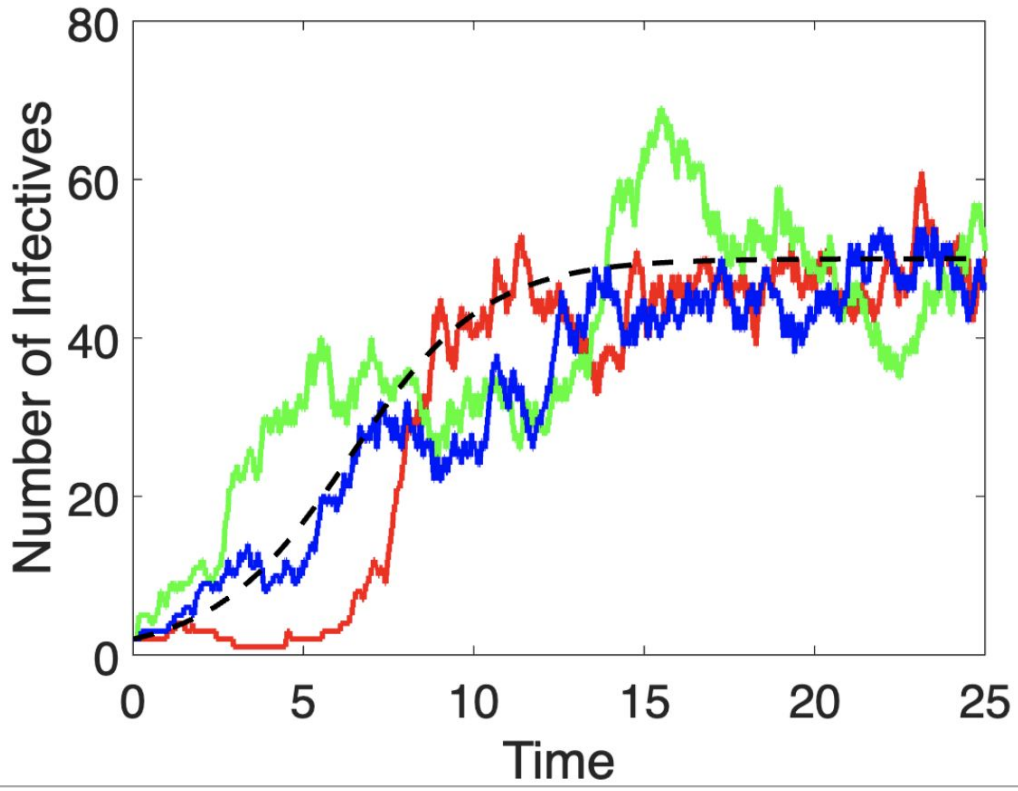
Following is the algorithm :-

- We define the constants that are $\beta, \gamma, \alpha, time, dt, N$ or total population.
- We begin a loop which goes from j= 1 to 3 (as we sample 3 paths) and initiate $i(1)$ or the number of infected people at 1st timestamp as some constant.
- In this loop we initiate another loop which goes from t=1 to $\frac{time}{dt}$
    1. We define two variables birth and death and calculate their value according to the formula for probability of a new infection and probability of a death or recovery respectively and the number of infected is taken as $i(t)$.
    2. And then we sample a random variable according to the distribution of the transition probabilities that were calculated in the previous step.

3. And if it is a birth process then $i(t+1) = i(t) + 1$.
4. And if it is a death process then $i(t+1) = i(t) - 1$.
5. And else $i(t+1) = i(t)$

- Then we plot the graph of i (infected) with the timestamps.

As an example here we simulate the dtmc SIS process on MATLAB where α=0.25, β=1, ɣ=0.25, N=100 and I(0)=2 (code for this given in the appendix).



4.3.2. Curve fitting

From the data that we have of the pneumococcus we try to find the value of the constants β, γ, α. We do this by finding the best fit of the deterministic SIS process curve on the data points that we already have of the disease.

We use the scipy library of python for doing this as it contains the curve-fit() function which allows us to do this.

Scipy is the scientific computing module of Python providing in-built functions on a lot of well-known Mathematical functions. The scipy.optimize package equips us with multiple optimization procedures.

Algorithm:-

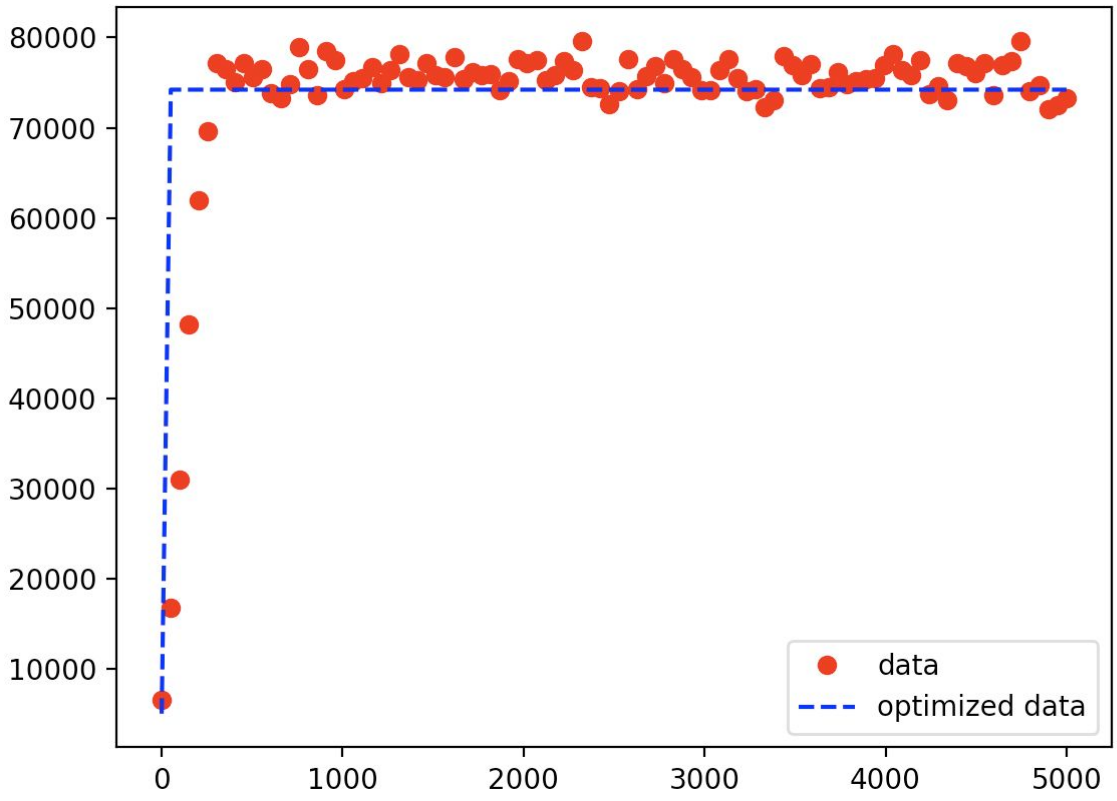
- We first store the value of the data points of pneumococcus in two variables 'i' for infectives and 't' for the timesteps.
- Then we define a function test that takes a time coordinate value from 't' , the constants β, γ, α and returns the value of number of infectives at that specific time coordinate according to the deterministic SIS model solution:-
- Then we implement the curve fit function which gives us the value of β, γ, α

$$I(t) = \begin{cases} \left[\left(\frac{\beta}{\beta N-\mu-\gamma}\right)\left(1-e^{-(\beta N-\mu-\gamma)t}\right)+\frac{e^{-(\beta N-\mu-\gamma)t}}{I(0)}\right]^{-1}; R_0 \neq 1 \\ \left[\beta t+\frac{1}{I(0)}\right]^{-1}; R_0=1 \end{cases}$$

that best fits the curve in the dataset distribution.

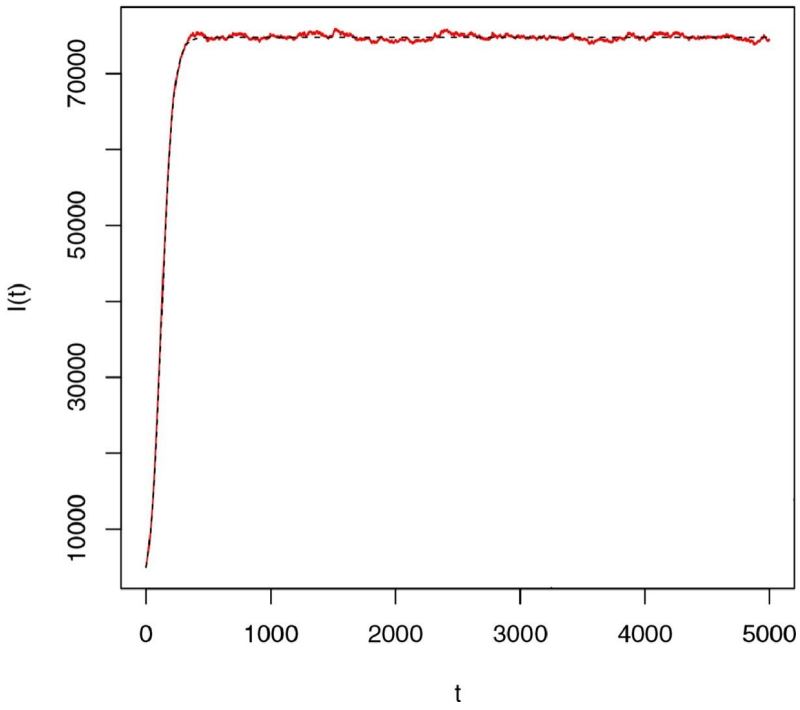
4.4. Results

We get the following result (graph and values of the constants) by doing curve fitting on the data of pneumococcus on python:-



And values of constants:-
α=0.00137363, β=0.0000002857, γ=0.02011
The code for this is given in the appendix.

On simulating the DTMC SIS process on MATLAB, we get the following simulation :-



The code for this is given in the appendix as well.

5. The SIR Model

5.1. The SIR Model Construction

The SIR Model consists of three compartments, Susceptible, Infected and Recovered. S(t), I(t), R(t) represent the number of susceptible, infected and removed individuals at time 't' respectively.

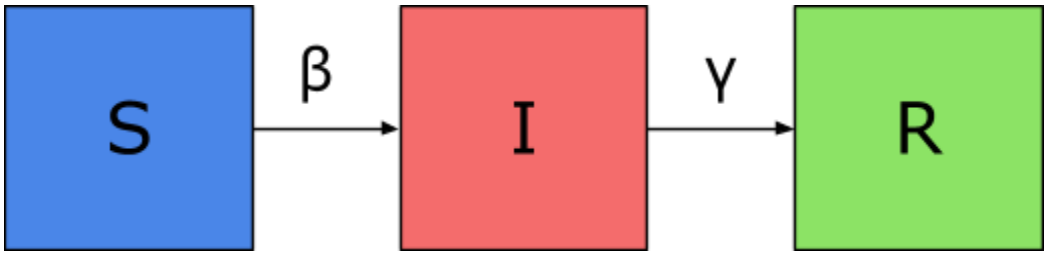
In this model, we would assume no births. There can only be infections, recovery and deaths as a result of infection. A susceptible individual can get infected by coming in contact with an infected individual. An infected individual may either die or recover from the disease. We would consider a dead and recovered person as the equal i.e. he would become a removed individual.

The total number of Susceptible, Infected and Recovered individuals in the system would remain constant i.e.

$$S(t) + I(t) + R(t) = N$$

where N is the total population size.

Now, we can proceed to define parameters for the disease. Let β be the Contact rate/Rate of transmission and γ be the Rate of removal. We will assume that there is a homogeneous interaction amongst the population. The figure given below provides a clearer understanding of the model.



A figure depicting the compartmental SIR Model

R_0 is known as the Basic Reproduction Number. It is defined as:

$$R_0 = \beta/\gamma$$

R_0 is used to determine the epidemic outcome when S(0)≈N. The following theorem can be used.

Theorem:
For the SIR Model,

- If $R_0\ S(0)/N \leq 1$, then disease-free equilibrium will be achieved and I(t) decreases monotonically to 0
- If $R_0\ S(0)/N > 1$, then an epidemic will occur and there will be an initial increase in I(t)

This is how an SIR Model is constructed. To implement the model, there are two broad approaches, a deterministic approach and a stochastic approach. These are discussed in detail in the coming sections.

5.2. The SIR Deterministic Model

A model in which there is no randomness involved in the development of the future states is called a deterministic model. It will always produce the same output from a given set of initial conditions. The initial conditions would be those defined above. The deterministic model can be defined using differential equations:

$$dS/dt = -\beta SI/N$$
$$dI/dt = \beta SI/N - \gamma I$$
$$dR/dt = \gamma I$$

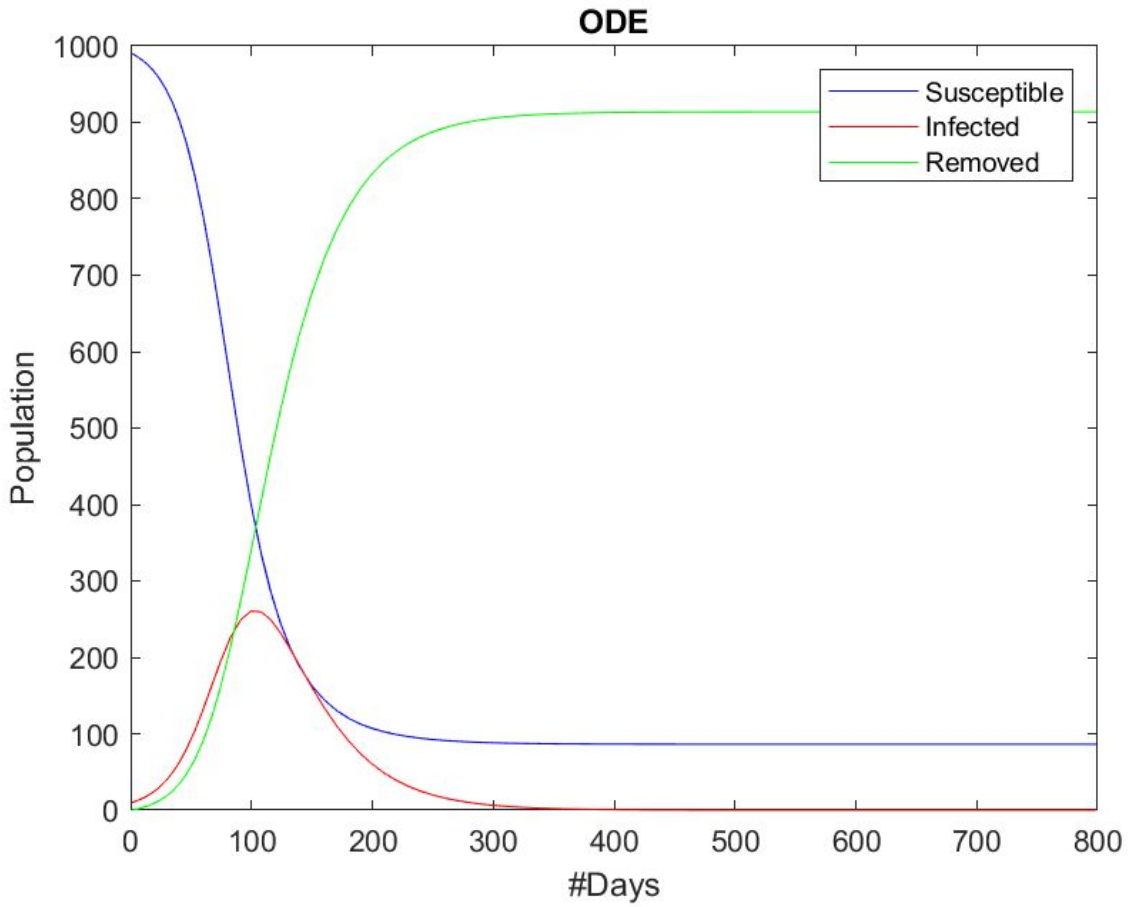
Since the total population is assumed to be constant, i.e. $S(t) + I(t) + R(t) = N$, these three equations can be reduced to two equations with two variables by taking $R(t) = N - S(t) - I(t)$. The two variables now are S(t) and I(t).

Parts of the script:-

odefunc(t, vars, beta, gamma): Sets ODE equations for SIR.

ode45: Non-stiff ODE solver in-built in MATLAB and most apt for this application.

det_ode(): Plots ODE



The Deterministic SIR Graph with N=1000, S(0)=990, I(0)=10, β =0.08, γ =0.03, Duration=800 days

5.3. The SIR Discrete Time Markov Chain (DTMC) Model:

S(t), I(t), R(t) represent the discrete random variables for no. of susceptible, infected and removed individuals at time 't' respectively
N is the total population.

Here $t \in \{0, \Delta t, 2\Delta t, ...\}$ and $S(t),\ I(t),\ R(t) \in \{0, 1, ..., N\}$. The variables t, S(t), I(t) and R(t) are all discrete. The remaining definitions and assumptions are similar to the deterministic model.
This is a bivariate process since there are 2 independent random variables, S(t) and I(t).
Notation- (s,i): State of the system when S(t)=s, I(t)=i and R(t)=N-S(t)-I(t)

The bivariate process $\{(S(t),\ I(t))\}$ at $t \in [0, \infty)$ has a joint probability function given by $p_{(s,i)}(t) = P[S(t) = s,\ I(t) = i]$
This process has the Markovian property and is time-homogeneous. To define the Markov chain, we can choose Δt to be sufficiently small such that at most one change in state can occur in the time Δt i.e. either there can be a new infection or a new death/recovery. The transition probability can be defined as:

$$\Delta p_{\Delta s,\Delta i} = p_{(s+\Delta s, i+\Delta i),\ (s,i)}(t) = P[S(t+\Delta t) = s + \Delta s,\ I(t) = i + \Delta i \mid S(t) = s, I(t) = i]$$

Here, $\Delta s \in \{-1, 0, 1\}$ and $\Delta i \in \{-1, 0, 1\}$

P(t)=$[\Delta p_{\Delta s,\Delta i}(t)]_{3x3}$ is a 3x3 Transition matrix defined as follows:

$$P(t) = \begin{bmatrix} o(\Delta t) & o(\Delta t) & \beta.si/N.\Delta t + o(\Delta t) \\ \gamma.i.\Delta t + o(\Delta t) & 1-(\beta.si/N+\gamma.i).\Delta t + o(\Delta t) & o(\Delta t) \\ o(\Delta t) & o(\Delta t) & o(\Delta t) \end{bmatrix}$$

Δt must be chosen such that probabilities of events lie in [0,1]
It is important to note that $(s, 0)$ are the absorbing states and $(s, i)\ s.t.\ i \neq 0$ are the transient states. The epidemic comes to an end when the absorbing state is reached.

5.4. The SIR Continuous Time Markov Chain (CTMC) Model:

Similar to the DTMC SIR model, $S(t),\ I(t) \in \{0, 1, 2, 3, ..., N\}$ but $t \in [0, \infty)$.
The remaining definitions and assumptions are similar to those described in the previous sections.
The transition probabilities associated with the stochastic process are defined for a small period of time Δt >0.

$$p_{(s,i),(s+k,i+j)}(\Delta t) = P[S(t+\Delta t) = s+k,\ I(t+\Delta t) = i+j | S(t) = s, I(t) = i]$$

The transition probabilities depend on the time between events Dt but not on the specific time t, a time-homogeneous process. In addition, given the current state of the process at time t, the future state of the process at time t þ Dt, for any Dt >0, does not depend on times prior to t, known as the Markov property
The Kolmogorov differential equations

The transition probabilities do not depend on a specific time 't'. Rather, it depends on Δt. The Markov property is also followed here i.e. given the current state at time 't', the future state at time t+ Δt does not depend on any time prior to 't'.

The transition probabilities can be defined as follows:

$p_{(s,i),(s+k,\ i+j)}(\Delta t) =$

□Is Δt/N + o(Δt)	(k,j)=(-1, +1)
γI Δt + o(Δt)	(k,j)=(0,-1)
1-(□Is/N+ γI) Δt +o(Δt)	(k,j)=(0,0)
o(Δt)	otherwise

$$p_{(s,i),(s+k,i+j)}(\Delta t) =$$

$$\sum_{k,j} P[S(t+\Delta t) = s+k,\ I(t+\Delta t) = i+j |\ S(t) = s, I(t) = i]\ \times P[S(t) = s, I(t) = i]$$

Dividing $p_{(s,i),(s+i,j+k)}(\Delta t)$ by Δt and letting $\Delta t \to 0$
Let a be the state at time 't'. If we want to reach state b=(s,i) at time t+ Δt, we get the following equation:

$$p_{a,(s,i)}(t+\Delta t) =$$
$$(\beta(s+1)(i-1)\Delta t/N)p_{a,(s+1,i-1)}(t) + (\gamma(i+1)\Delta t)p_{a,(s,i+1)}(t) + (1-(\beta si/N+\gamma i)\Delta t)p_{a,(s,i)}(t) + o(\Delta t)$$

Now, on subtracting $p_{a,(s,i)}(t)$ and taking $\Delta t \to 0$, we get the forward Kolmogorov equation.

$$dp_{a,(s,i)}(t)/dt =$$
$$(\beta(s+1)(i-1)/N)p_{a,(s+1,i-1)}(t) + (\gamma(i+1))p_{a,(s,i+1)}(t) - (\beta si/N+\gamma i))p_{a,(s,i)}(t)$$

The forward equations are used to predict future dynamics and are known as the "master equations".

We can use a similar approach to get the backward Kolmogorov equations. Let the initial state be a=(s,i). In the time Δt, an event can occur and for the remaining time, there is a transition to state b.

$$p_{(s,i),b}(t+\Delta t) =$$
$$(\beta si\Delta t/N)p_{(s-1,i+1),b}(t) + (\gamma i\Delta t)p_{(s,i-1),b}(t) + (1-(\beta si/N+\gamma i)\Delta t)p_{(s,i),b}(t) + o(\Delta t)$$

Now, on subtracting $p_{(s,i),b}(t)$ and taking $\Delta t \to 0$, we get the backward Kolmogorov equation. The backward equation is used to study the end of the epidemic.

$$dp_{(s,i),b}(t)/dt = (\beta si/N)p_{(s-1,i+1),b}(t) + (\gamma i)p_{(s,i-1),b}(t) - (\beta si/N+\gamma i)p_{(s,i),b}(t)$$

We can define an infinitesimal matrix $Q_{(N+1)(N+2)/2}$ for states (s,i).

5.4.1. The Branching Process Approximation

The branching process is used to estimate the probability of a minor outbreak for the CTMC model near disease free equilibrium. We study the stochastic behaviour near disease free equilibrium when a few infectious individuals have been introduced to the population. Techniques of probability generating functions (pgfs) are also widely used here.

Initially, $S(t) \approx N \Rightarrow$ Infection rate = $\beta.si/N \approx \beta i$ and

Removal rate = γi

Assumptions:

- Each infectious individual's behavior is independent from other infectious individuals.
- Each infectious individual has the same probability of recovery and the same probability of transmitting an infection.
- The susceptible population is sufficiently large.

We will use two PGFs to study the probability of extinction. The offspring pgf applies to each infectious individual as defined below:

$f(u) = \sum_{j=0}^{\infty} p_j u^j$ where $u \in [0, 1]$

where p_j=probability of one individual generating j new individuals of the same type

$$f(u) = (\gamma + \beta u^2)/(\beta + \gamma)$$

Properties of the pgf:

$$f(1) = 1$$

$$f'(1) = E(j)$$

In f(u), the first term is the probability that an infectious individual recovers whereas the coefficient of the second term is the probability that an infectious individual infects another individual. The power of u determines the number of infectious individuals generated from one infectious individual. The mean value of the above equation is:

$$E(j) = f'(1) = 2\beta/(\beta + \gamma)$$

This acts as a threshold parameter but is different from the Reproduction number R_0.

It is important to note that:

- $f'(1) \neq R_0$
- $f'(1) > 1 \Leftrightarrow R_0 > 1$

Proof:

$$R_0 = \beta/\gamma$$

$$f'(1) = 2\beta/(\beta + \gamma)$$

$$1/R_0 = \gamma/\beta$$

$$1/f'(1) = (\beta + \gamma)/2\beta = (1 + 1/R_0)/2$$

$$f'(1) = 2(1 - 1/(R_0 + 1))$$

$$R_0 > 1 \Rightarrow 1/(R_0+1) < 1/2 \Rightarrow 1 - 1/(R_0+1) > 1/2 \Rightarrow f'(1) > 1$$
$$f'(1) > 1 \Rightarrow 1 - 1/(R_0+1) > 1/2 \Rightarrow 1/(R_0+1) < 1/2 \Rightarrow R_0 > 1$$

Since this is a branching process, a fixed point of offspring pgf gives the asymptotic probability of extinction.

Now, we will show that the fixed points of f are the stationary/time-independent solutions of the branching process approximation for the probability of extinction of the infectious class I(t).

SIR Branching Approximation PGF:

G_i: pgf for the infectious class I(t), given I(0)= i
$p_{i,j}(t)$: P[I(t)=j|I(0)=i]

$$G_i(u,t) = E(u^{I(t)}|I(0)=i) = \sum_{j=0}^{\infty} p_{i,j}(t)u^j$$

For the branching approximation of I(t), the Backward Kolmogorov differential equations are:

$$dp_{i,k}(t)/dt = \beta i p_{i+1,k}(t) + \gamma i p_{i-1,k}(t) - (\beta+\gamma) i p_{i,k}(t)$$

For k=0,

$$dp_{i,0}(t)/dt = \beta i p_{i+1,0}(t) + \gamma i p_{i-1,0}(t) - (\beta+\gamma) i p_{i,0}(t)$$

$G_i(0,t)$ denotes probability of disease extinction by time t

$$G_i(0,t) = p_{i,0}(t)$$
$$\partial G_i(0,t)/\partial t = \beta i G_{i+1}(0,t) + \gamma i G_{i-1}(0,t) - (\beta+\gamma) i G_i(0,t)$$

Assume each individual's independent behaviour

$$G_i(u,t) = (G_1(u,t))^i$$
$$\partial G_i/\partial t = i {G_1}^{i-1} \partial G_1/\partial t$$
$$i {G_1}^{i-1} \partial G_1/\partial t = \beta i {G_1}^{i+1} + \gamma i {G_1}^{i-1} - (\beta+\gamma) i {G_1}^{i}$$

f is the offspring's pgf

$$\partial G_1/\partial t = \beta {G_1}^2 + \gamma - (\beta+\gamma) G_1 = (\beta+\gamma)\,[f(G_1) - G_1]$$

$$\partial p_{1,0}(t)/\partial t = (\beta+\gamma)\,[f(p_{1,0}(t)) - p_{1,0}(t)]$$

The steady state values of this equation are the stationary solutions for this process.

$\partial p_{1,0}(t)/\partial t = 0$

$f(u) = (\gamma + \beta u^2)/(\beta + \gamma) = u$

$\beta u^2 - (\beta + \gamma)u + \gamma = 0$

$(\beta u - \gamma)(u - 1) = 0$

$u = \gamma/\beta$ or $u = 1$

These are the possible steady state values

$\partial^2 p_{1,0}(t)/\partial t^2 < 0$

$f'(u) - 1 = 2\beta u/(\beta + \gamma) - 1 < 0$ for stable steady state

$\Rightarrow \gamma/\beta$ when $2\gamma/(\beta + \gamma) < 1$ or 1 when $2\beta/(\beta + \gamma) < 1$

$\Rightarrow 1/R_0$ is stable when $R_0 > 1$ or 1 is stable when $R_0 < 1$

For $I(t) = i > 1$, by assumption of independent infectious individuals

Asymptotic steady state values are $(1/R_0)^i$ when $R_0 > 1$ or 1 when $R_0 < 1$

$$P_{\text{minor outbreak}} = \begin{cases} (1/R_0)^i & R_0 > 1 \\ 1 & R_0 < 1 \end{cases}$$

$$P_{\text{major outbreak}} = \begin{cases} 1-(1/R_0)^i & R_0 > 1 \\ 0 & R_0 < 1 \end{cases}$$

Since these estimates are approximations of the branching process, a large population with few infectious individuals will yield a higher accuracy.

5.4.2. Gillespie Algorithm

From the CTMC SIR model, we obtain Kolmogorov differential equations. This model is reduced from three variables to two random variables. This is now a bivariate process. Solving the Kolmogorov equation for a bivariate process is very difficult. Hence, it is better to numerically simulate stochastic realizations of the process. To

simulate such processes, the gillespie algorithm is widely used. This algorithm generates a statistically correct trajectory of a stochastic equation.

We will take two uniformly random numbers $u_1, u_2 \in U[0,1]$

u_1 is used to simulate intervent time
The Markov property implies intervent time T has an exponential distribution i.e.
$T \sim \lambda e^{-\lambda t}$
where λ is the sum of rates of all possible events
$\lambda = \beta si/N + \gamma i$
where (s,i) are the state of the system at time 't'

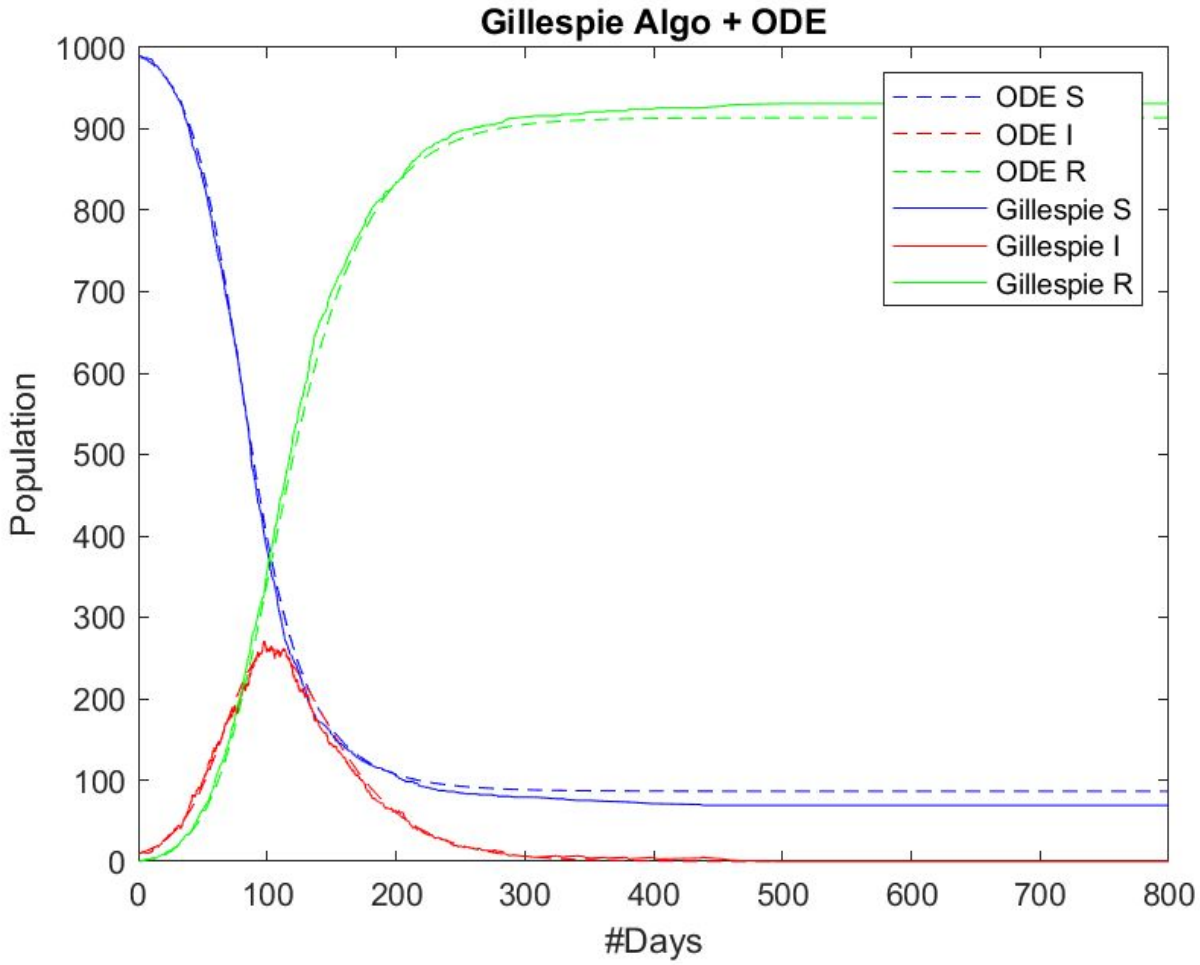
F(t): Cumulative probability=P[T≤t]
$F(t) = 1 - e^{-\lambda t}$
Let $u_1 = P[T > t] = 1 - F(t) = e^{-\lambda t}$
$\Rightarrow \; t = - ln(u_1)/\lambda$

u_2 is used to simulate occurrence of particular event
$p_1 = (\beta si/N)/\lambda \qquad p_2 = (\gamma i)/\lambda$
$p_1 + p_2 = 1$
Now, we can split interval [0,1] into $[0,p_1]$ and $(p_1,1]$
If $u_2 \in [0,p_1]$ event 1 occurs, i.e. $(s,i) \rightarrow (s-1, i+1)$
If $u_2 \in (p_1,1]$ event 2 occurs, i.e. $(s,i) \rightarrow (s, i-1)$
This is the implementation of the Gillespie Algorithm to simulate the SIR model.

The graph below shows the Gillespie Algorithm along with the deterministic SIR graph. We can see the difference between a simulated model and a stochastic model.

Parts of the script:-

gillespie(): Simulates the epidemic stochastically according to the Gillespie algorithm and plots it.



The Gillespie Algorithm Vs. The Deterministic model
with N=1000, S(0)=990, I(0)=10, β =0.08, γ =0.03, Duration=800 days

5.5. SIR Stochastic Differential Equations

Stochastic differential equations for the SIR epidemic model follow from a diffusion process. The random variables are-
S(t), I(t), R(t) which are continuous RVs for susceptible, infected and removed populations.
Also- $S(t),\ I(t),\ R(t) \in [0, N]$

Remaining definitions and assumptions are similar to CTMC.

The SDE's are modelled as a diffusion process. They are useful in simulation of sample paths of infectious-disease spread over large populations. The SDEs are also used as they are easier in simulating than the Kolmogorov Equations.

5.5.1. Derivation Of SDE

To derive the Stochastic Differential Equations we use a heuristic approach.

The time interval [0,t] is divided into subintervals of length Δt.

Let $\Delta X(t) = \begin{bmatrix} \Delta S(t) \\ \Delta I(t) \end{bmatrix}$

Δt is subdivided further into $\Delta t_i = t_i - t_{i-1}$ where i=1,...,n; $t_0 = t$, $t_n = t + \Delta t$

$$\sum_{i=1}^{n} \Delta t_i = \Delta t$$

$$\Delta X(t) = \sum_{i=1}^{n} \Delta X(t_i)$$

For sufficiently small enough Δt_i we can reasonably assume $\{\Delta X(t_i)\}$ on Δt are independent and identically distributed.

For sufficiently large n,
The Central Limit Theorem -
$\Delta X(t) - E[\Delta X(t)] \sim Normal(0, CV[\Delta X(t)])$,where 0 is the zero vector

$$E[\Delta X] = \begin{bmatrix} -\beta SI/N \\ \beta SI/N - \gamma I \end{bmatrix} \Delta t$$

$$CV[\Delta X] \approx E[\Delta X^T \Delta X] = E\begin{bmatrix} (\Delta S)^2 & \Delta S \Delta I \\ \Delta I \Delta S & (\Delta I)^2 \end{bmatrix}$$

$$= \begin{bmatrix} \beta SI/N & -\beta SI/N \\ -\beta SI/N & \beta SI/N + \gamma I \end{bmatrix} \Delta t$$

$f\Delta t$: $E[\Delta X]$
$C\Delta t$: $CV[\Delta X]$

W(t) : vector of 2 independent Wiener processes
$\Rightarrow W_i(t) \sim Normal(0, t)$

$$W(t) = \begin{bmatrix} W_1(t) \\ W_2(t) \end{bmatrix}$$

$$\Delta W(t) = \begin{bmatrix} \Delta W_1(t) \\ \Delta W_2(t) \end{bmatrix}$$

To write the Stochastic Differential Equations of the SIR stochastic process we need the square root of the covariance matrix $C\Delta t$. Therefore, we find matrix G s.t. G is the sq. root of $CV[\Delta X]$ or $G^TG=C$
Note - G may not be unique

$$G = \begin{bmatrix} -\sqrt{\beta SI/N} & \sqrt{\beta SI/N} \\ \sqrt{\beta SI/N} & -\sqrt{\gamma I} \end{bmatrix}$$

Therefore,
$$\Delta X(t) \approx f(X(t))\Delta t + G(X(t))\Delta W(t)$$

Now as $\Delta t \to 0$,
$$dX(t) = f(X(t))dt + G(X(t))dW(t)$$
where, $dW_i(t) \approx W_i(t + dt) - W_i(t) \sim Normal(0, dt)$

This stochastic differential equation is known as an Itô SDE because the right side is evaluated at time t

Writing the rows of the matrix separately we get,
$$dS = -[\beta SI/N]dt - \sqrt{\beta SI/N}\, dW_1(t)$$
$$dI = [\beta SI/N - \gamma I]dt + \sqrt{\beta SI/N}\, dW_1(t) - \sqrt{\gamma I}\, dW_2(t)$$

If Wiener processes are neglected, a deterministic model is obtained.

5.5.2. Euler-Maruyama method

The EM method is a numerical method to stipulate sample path for SDE model
Finite difference approximation
$t = 0, \Delta t, 2\Delta t, ...$

η_i:Independent standard normal random numbers
$\eta_i \in Normal(0, 1)$

Since in our case there are 2 Wiener processes, we take two values of η.

$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$

Therefore we have,
$X(t + \Delta t) \approx X(t) + \Delta X(t) = X(t) + f(X(t), t)\Delta t + G(X(t), t)\eta\sqrt{\Delta t}$

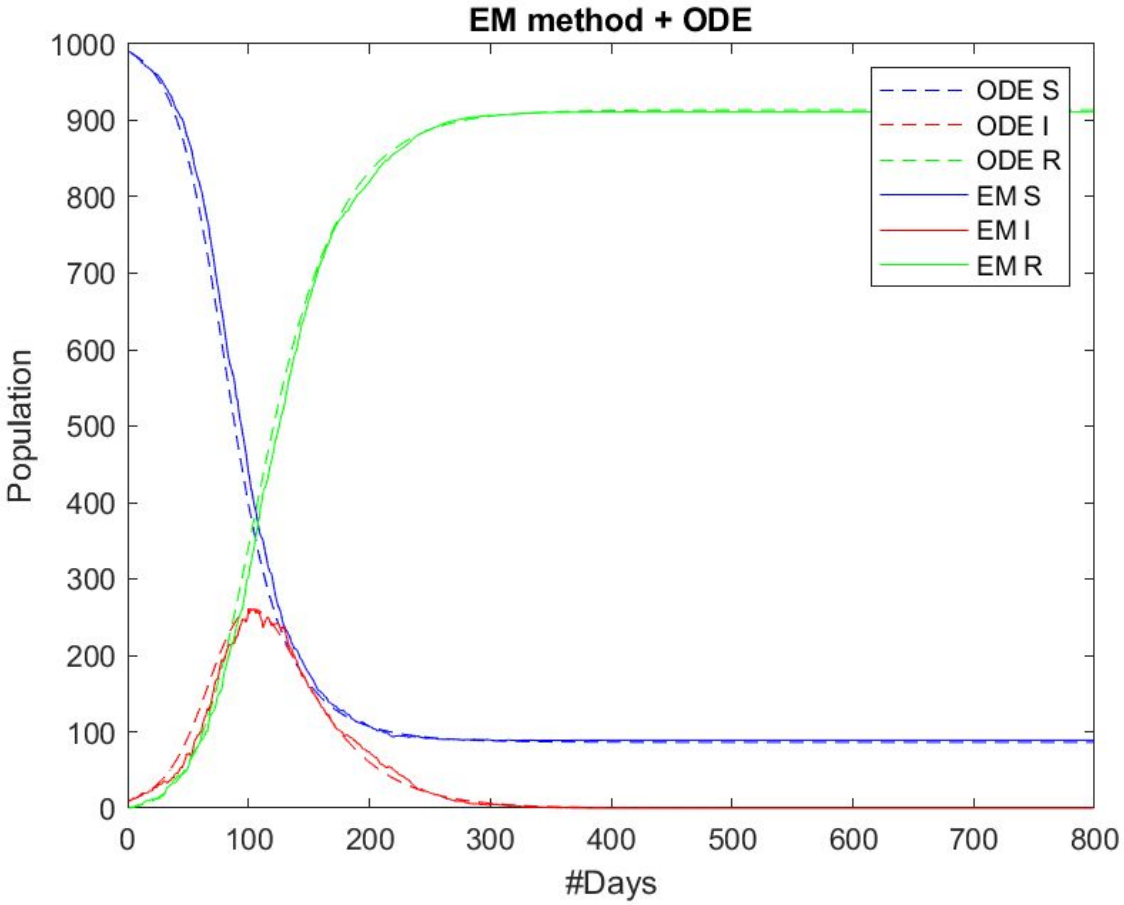
Δt is chosen to be sufficiently small enough to ensure convergence.

The sample paths which are obtained are continuous but not differentiable, as is the property of Wiener process.

This simulation method ensures faster and simpler simulation than computing sample paths for CTMC model for large populations. This is because in SDE models the time step, even though small, has a fixed length. Whereas in the CTMC case the intervent time τ has to be calculated for each change in ΔX and it decreases as the population increases which is the case here.

Parts of the script:-

EM_method(): Simulates the epidemic stochastically according to the EM-method and plots it.



The EM method Vs. The Deterministic model

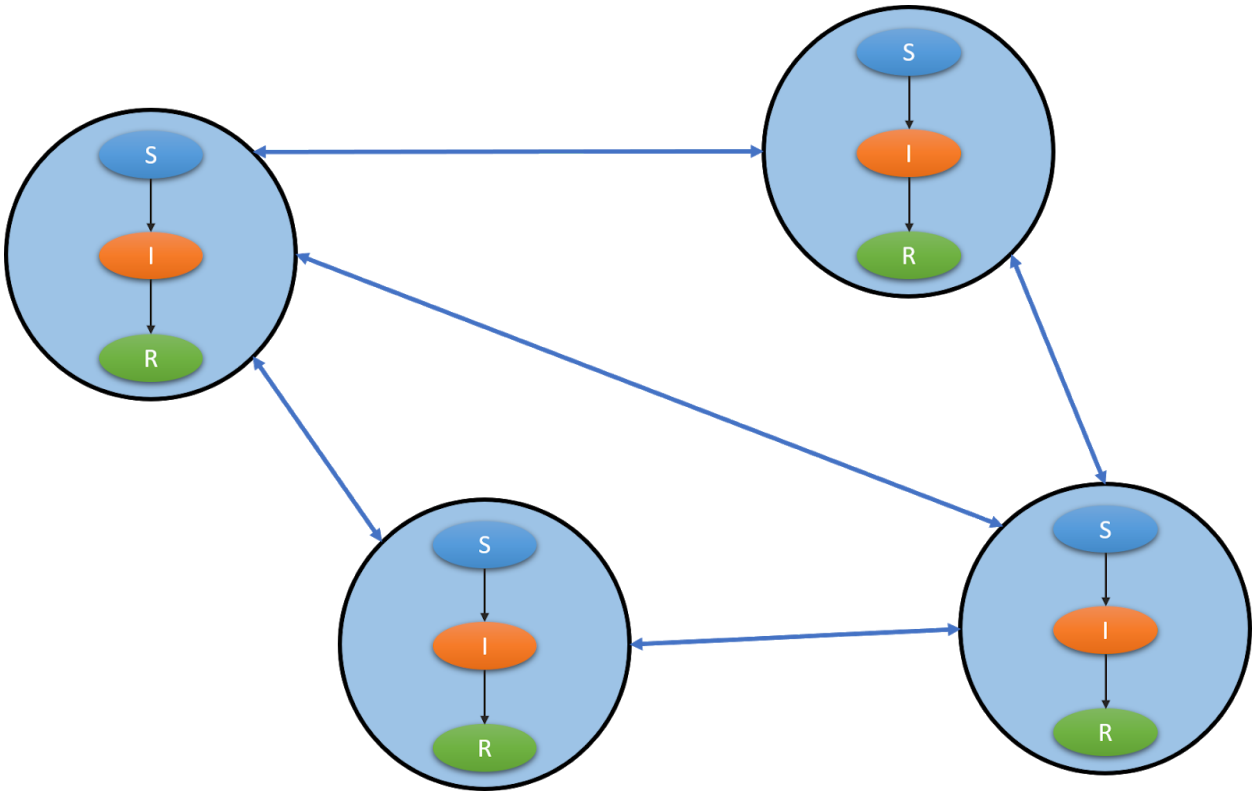
with N=1000, S(0)=990, I(0)=10, β =0.08, γ =0.03, Duration=800 days

5.6. The SIR Model for a Metapopulation

A group of populations that are separated by space but consist of the same species is called a metapopulation. These spatially separated populations interact as individual members move from one population to another.

Here, we will have a look at the spread of a disease that follows a SIR model, amongst different clusters of a population.

We can view the collection of populations as a network. The populations are represented by nodes and the interaction between the regions will be the edges of the network. We can make a variety of assumptions about the connecting edges. One can assume that people in each region are interacting with people in every other region. Hence, this can be represented as a digraph. A digraph with 4 clusters of population is shown below:



A digraph showing 4 clusters of population that have been infected with an SIR disease. The arrows depict the connection between the two clusters

For the metapopulation model, there are two broad ways for specifying the interaction, the cross coupled model and the mobility model.

5.6.1 The Cross Coupled Model

The cross coupled model assumes that the different regions are connected. Each population cluster will have its own set of SIR equations. We can index each set of equations by 'i' where 'i' represents the SIR equations for the i^{th} region. Let there be 'm' clusters of population.

$$dS_i/dt = -\beta_i S_i I_i/N$$
$$dI_i/dt = \beta_i S_i I_i/N - \gamma I_i$$
$$dR_i/dt = \gamma I_i$$

$$where\ i \in \{0, 1, ..m\}$$

Now, we need to consider that an infection in one region may be caused by an infective from another region, we write β with two indices, i.e. β_{ik}

β_{ik} represents the contact rate between susceptible of region i and infectives of region k. It would now be required to sum across all the m regions as shown below:

$$dS_i/dt = -S_i \sum_{k=1}^{m} \beta_{ik} I_k/N$$
$$dI_i/dt = S_i \sum_{k=1}^{m} \beta_{ik} I_k/N - \gamma I_i$$
$$dR_i/dt = \gamma I_i$$

We can write β in a matrix form as

$$\begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \beta_{m1} & \beta_{m2} & \dots & \beta_{mm} \end{bmatrix}$$

Here, the number of parameters are $m \times m$ which is very large. This matrix can be simplified. The matrix is symmetric so now there will be $m \times m/2$ parameters. For the clusters that are not interacting, their entry in the matrix will be zero.

Therefore in the cross coupled model, the individuals are identified within clusters and they don't change clusters. But there are some contacts with other clusters that cause the transmission between the clusters. This is the cross coupled SIR model

5.6.2. The Mobility Model

The mobility model assumes that there is interaction between clusters that happen due to movement of people. Both susceptible as well as infected individuals move around across clusters.

Let θ_{ik} represent the proportion of people who travel from cluster k to cluster i. Many assumptions can be made according to the requirements. For example, θ_{ik} can represent only the infected people or it can represent both, the infected as well as susceptible.

The equations will now become

$$dS_i/dt = -S_i\beta_i I_i/N + \sum_{k=1}^{m} \theta_{ik} S_k$$

$$dI_i/dt = S_i\beta_i I_i/N - \gamma I_i + \sum_{k=1}^{m} \theta_{ik} S_k$$

$$dR_i/dt = \gamma I_i$$

In determining the level of interactions between clusters, we can also use parametric functions. For example, when talking about interaction between two clusters, we can use the gravity model. The masses of the planets are analogous to the populations of clusters.

β_{ij} can be represented as:

$$\beta_{ij} = G \frac{N_i^a N_j^b}{f(d_{ij})}$$

$$where\ N_i : population\ of\ cluster\ i$$

$$f(d_{ij}) : function\ of\ distance\ between\ cluster\ i\ and\ j$$

$$a, b : \ arbitrary\ constants$$

The radiation model is also widely used:

$$\beta_{ij} = C_j \frac{N_i N_j}{(N_i + P_{ij})(N_i + N_j + P_{ij})}$$

$$where\ C_j :\ no.\ of\ people\ who\ move\ from\ cluster\ j$$

$$P_{ij} : no.\ of\ people\ who\ live\ between\ cluster\ i\ and\ j\ excluding\ clusters\ i\ and\ j$$

Another model that is used is based on the radiation law:

$$\beta_{ij} = (1 + \frac{d_{ij}}{a})^{-b}$$

$$where\ d_{ij} : distance\ between\ cluster\ i\ and\ j$$

$$a, b : arbitrary\ constants$$

These three parametric functions are usually used in the mobility model. The cross-coupled and mobility model are two broad categories of metapopulation modelling.
To better fit the Coronavirus, we can make some changes that would improve accuracy . We can expect that the interaction amongst people happens via daily commute of people or other travels of similar nature.
Now in each region, there are residents and people from other clusters. We can represent the number of susceptibles, infectives, and recovered by two indices.
S_{ik} represents the number of susceptibles who are residents of cluster i, but are visiting cluster k
S_{ii} represents the number of susceptibles that are in cluster i and belong to cluster i itself

Similarly, we can represent Infectives and recovereds.
There are m regions in total, then from the region i perspective, there are m-1 other regions from which people can visit, so there will be m-1 sets of equations for the visitors, but we can reduce it to only one set for a generic cluster k.

Now we will form the differential equations. We will assume that the contact rate is the same within the same cluster for both locals and visitors.
Let us represent the rate at which people leave cluster i and go to cluster k by l_{ik}
and the people who return to cluster i from cluster k by r_{ik}
The equations for cluster i are as follows:

$$dS_{ii}/dt = -S_{ii}\beta_i \sum_{k=1}^{m} I_{ki}/N_i - \sum_{k=1}^{m} S_{ii}l_{ik} + \sum_{k=1}^{m} S_{ik}r_{ik}$$

$$dS_{ki}/dt = \; -S_{ki}\beta_i \sum_{k=1}^{m} I_{ki}/N_i + S_{kk}l_{ki} - S_{ki}r_{ki}$$

For a better representation, we can denote the force of infections as

$$\lambda_i = \beta_i \sum_{k=1}^{m} I_{ki} / N_i$$

Our equations now become:

$$dS_{ii}/dt = \; -S_{ii}\lambda_i - \sum_{k=1}^{m} S_{ii}l_{ik} + \sum_{k=1}^{m} S_{ik}r_{ik}$$

$$dS_{ki}/dt = \; -S_{ki}\lambda_i + S_{kk}l_{ki} - S_{ki}r_{ki}$$

Using a similar approach, we can define the equations for I and R.
These equations can be simulated in a stochastic manner and interesting conclusions can be drawn from this model.Since there are many clusters and variables, simulation of such a model requires the use of computers with large processing power.

5.7. Estimation Of Parameters

The estimation of parameters is done by minimising the error between the predicted values and the real values. The assumptions of the model are – a constant population, uniform mixing of people as well as people moving to the recovered stage being equally likely.

The model relies heavily on data and hence its forecast depends on it.

For estimating the parameters, they are obtained by minimizing an objective function which is the sum of squares of the differences in the predicted and real values. This is achieved using the **fminsearch** function from the optimization toolbox.

The fminsearch function -

It is a nonlinear programming solver. Searches for the minimum of a problem specified by min $f(x)$ where $f(x)$ is a function that returns a scalar, and x is a vector or a matrix.

It returns the value of x for which the function is at its minimum as well as the minimum value.

fminsearch uses the Nelder-Mead simplex algorithm.

Parts of the script:-

parest(): This function is used to call the fminsearch function from the optimization toolbox for minimizing the error with the help of the optimization function i.e. optim_fun. The first value of b0 i.e the parameters is decided by the iniGuess function here.b0 is the initial guess for beta and gamma which serves as the initial point for the fminsearch function.

optim_fun(params): This function is where the mean squared error is defined. This optimization function is then used for estimation with the fminsearch optimizer.

This function takes input params which consists of the values of beta and gamma which come from each iteration of the fminsearch function.

odefunc(t, vars, beta, gamma): This function is used to define the SIR model, which is then solved by the ode45 solves available in MATLAB with the recent beta

and gamma values. This returns the values of S,I,R which are now the predicted values for the next iteration of fminsearch.

This function takes input t, vars and beta and gamma.T is the time, vars consists of susceptible, infected, and recovered values.

iniGuess(): This function is used to set the values of beta and gamma to uniform random numbers as initial guesses.

5.8. COVID-19 in India - An Analysis

5.8.1. Data

The dataset used contains time series data of country-wise confirmed cases, reported deaths and reported recoveries for COVID-19 is maintained by John Hopkins University CSSE and is updated daily. It is licensed under the Open Data Commons Public Domain and Dedication License.

Using this, data pertaining to India is extracted and then converted to SIR data.

Parts of the script:-

data_extraction.py: Python code using Regular expressions to extract date, confirmed, dead and recovered data from original dataset **time-series-countries.txt** into **date-India-confirmed-recovered-dead.csv**.

get_data(): MATLAB code to initialize total population of India, import csv-file **date-India-confirmed-recovered-dead.csv** as a table and convert it into S,I and R data.

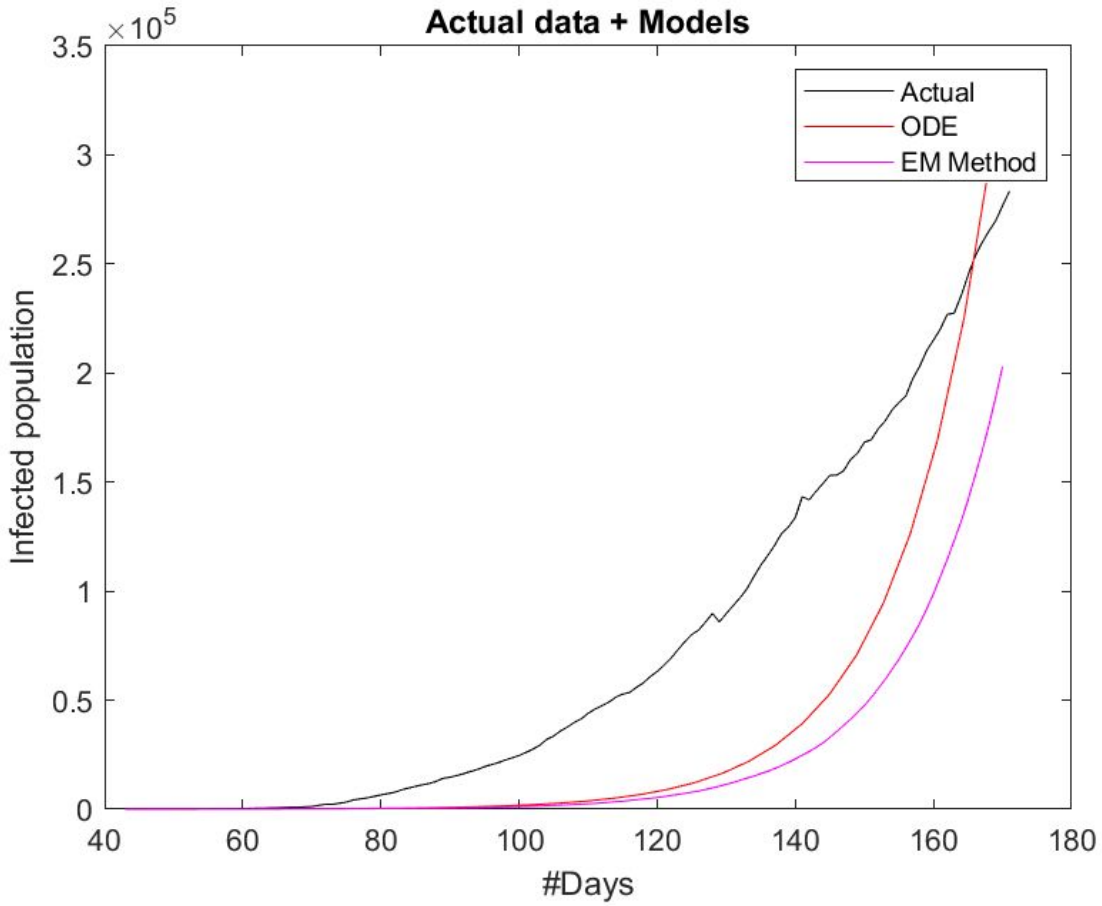
view_data(): MATLAB code for plotting actual data.

5.8.2. Overall Analysis

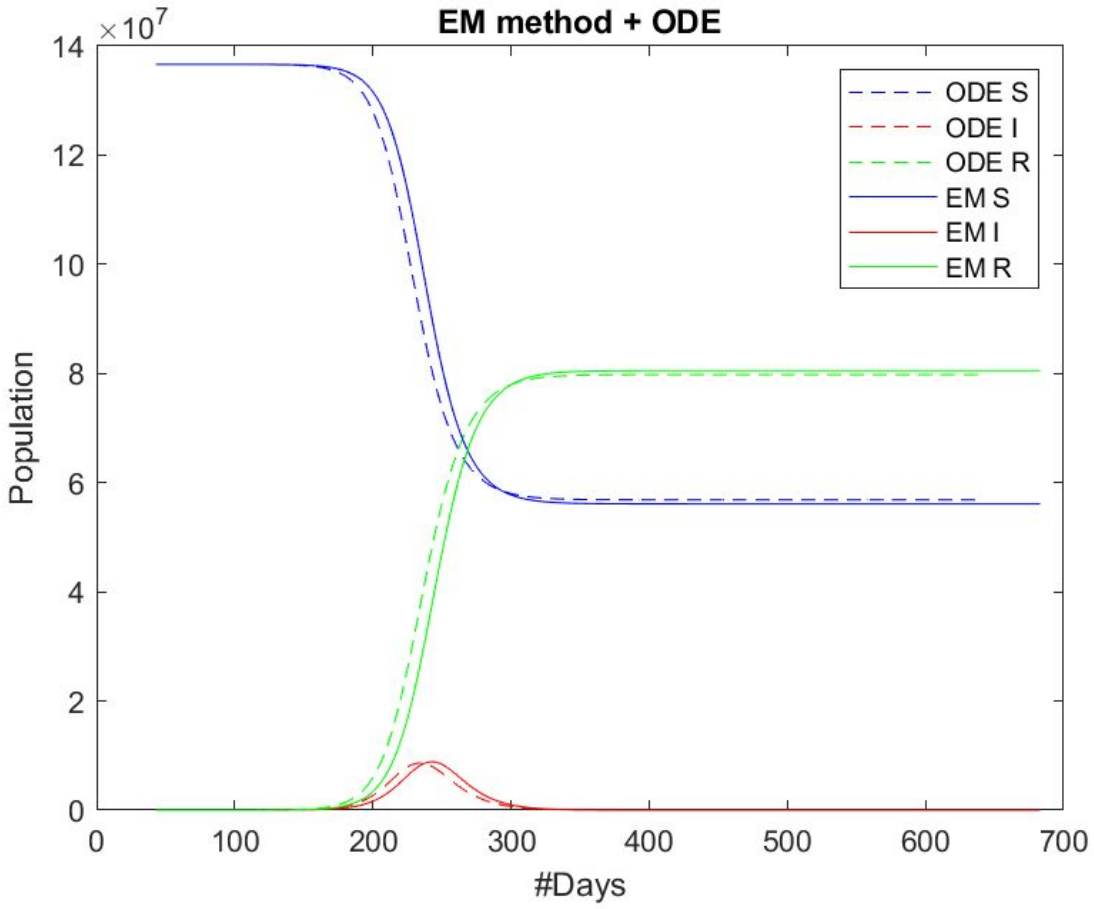
Initially, the entire time-span is considered, from the start of disease spread in India(4th March,2020) up to now (10th July,2020) for an elementary analysis. The parameters beta and gamma are estimated using actual data and then the deterministic and stochastic models for around 700 days from the start date were simulated and plotted.

```
COVID-19 in India - Overall Analysis
Total population=136641750    Initial Infectives=25    Basic Reproduction Number=1.5026
Start Date:2020-03-04    End Date:2020-07-10    Transmission Rate(Beta)=0.22454    Removal Rate(Gamma)=0.14943
Optimization status=1    Optimized error function=12584.0902
Probabilty of major outbreak=0.99996    Total simulated number of cases=79785782.8773
Deterministic Status: Epidemic
```

Properties obtained for overall analysis



Understanding accuracy of estimation and comparison of actual data, deterministic and stochastic models for overall analysis



Deterministic and stochastic simulation for longer duration for overall analysis

Since the number of infectives became quite high on the later dates (~o(10^5)), simulation using Gillespie algorithm was not feasible as it would need to run for ~o(10^5) iterations and store and plot that much data which heavily slowed down the program.

However, considering the lockdown conditions while analysing is necessary for better understanding and policy formulation for future action. Thus we have undertaken phase-wise analysis of the spread of COVID-19 in India in the next part.

5.8.3. Phase-wise Analysis

The timeline is split into 3 phases:

1. Pre-Lockdown
2. Lockdown
3. Post-Lockdown

This is done because according to theory, the transmission rate of the disease is directly proportional to the contact between infected individuals. Under lockdown, we assume there is significantly lesser contact between infected and susceptible populations due to isolation of detected infectives and social distancing norms. This should affect the transmission parameter beta. The changes in the future dynamics of disease-spread thus reflect the benefit or futility of taking such precautionary measures.

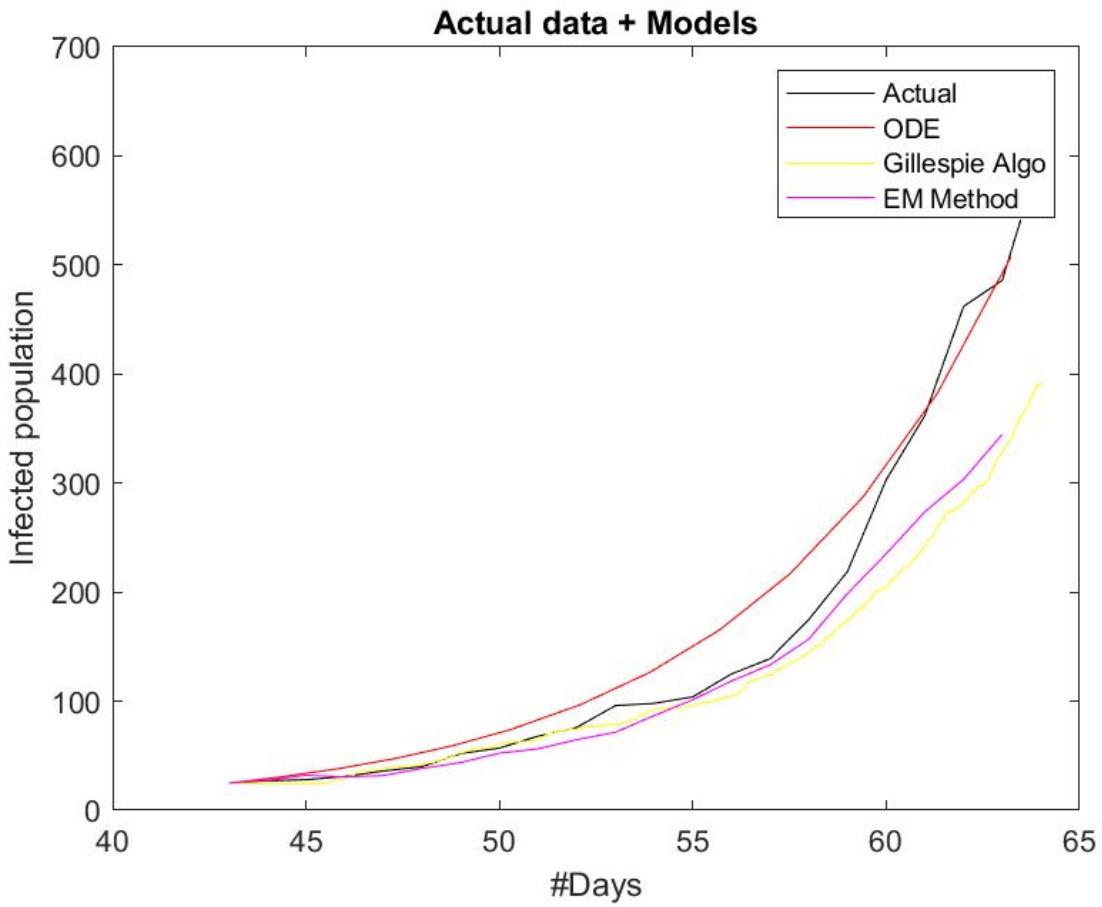
The parameters beta and gamma have been estimated for the duration of each phase. By plotting the actual data for the number of infectives during a particular phase against the simulations according to the ODE, EM-method and in some places the Gillespie algorithm we can observe the difference between the various methods. Also calculated are the probability of a major outbreak occurring, the status of whether an epidemic will occur or disease-free equilibrium according to the previously discussed theorem and the total number of people that suffered from the disease helping us understand the damage done at the end of the epidemic. The deterministic and stochastic plots using EM-method for a longer period of time are also included to observe the future dynamics if the beta and gamma were to remain unchanged.

5.8.3.1. Phase 1

In the initial phase of the disease in India (December 2019-February 2020), there were none to a few random cases detected, usually on foreign nationals. Only around the beginning of March, the disease significantly started spreading among Indians. Thus the first phase under observation here starts on 4th March 2020 and goes on up to the announcement of the nationwide lockdown on 25th April 2020.

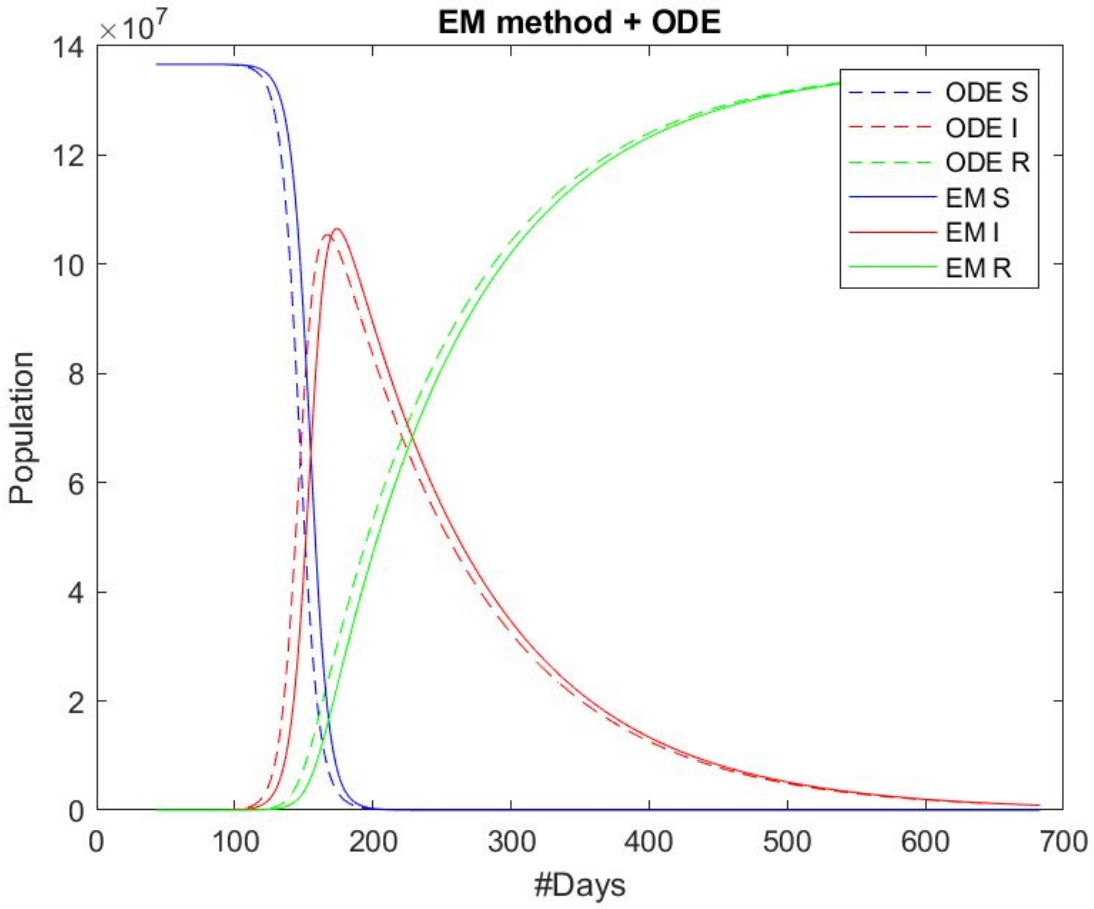
```
COVID-19 in India - Phase 1 Analysis - Pre-Lockdown
Total population=136641750    Initial Infectives=25    Basic Reproduction Number=16.7182
Start Date:2020-03-04    End Date:2020-03-25    Transmission Rate(Beta)=0.15832    Removal Rate(Gamma)=0.00947
Optimization status=1    Optimized error function=12.6782
Probabilty of major outbreak=1    Total simulated number of cases=135358634.1041
Deterministic Status: Epidemic
```

Properties obtained for phase-1 analysis



Understanding accuracy of estimation and comparison of actual data, deterministic and stochastic models for phase-1 analysis

Since the number of cases in this period wasn't extremely high we can also observe the Simulation using the Gillespie algorithm here.



Deterministic and stochastic simulation for longer duration for phase-1 analysis

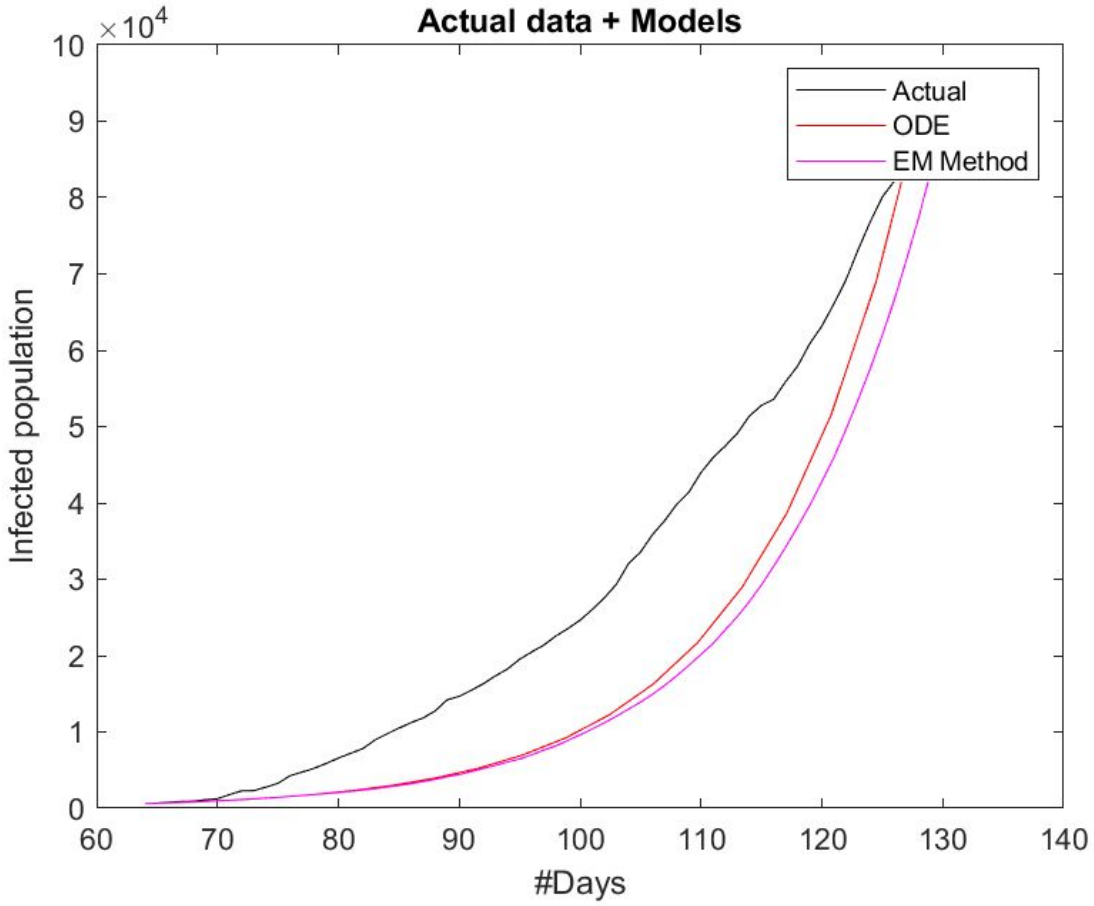
It can be easily observed that if no measures had been undertaken in the beginning the disease would have spread uncontrollably and eventually infected almost the entire population over a span of around two years. Taking necessary precautionary steps urgently right in the beginning was a smart calculated move.

5.8.3.2. Phase 2

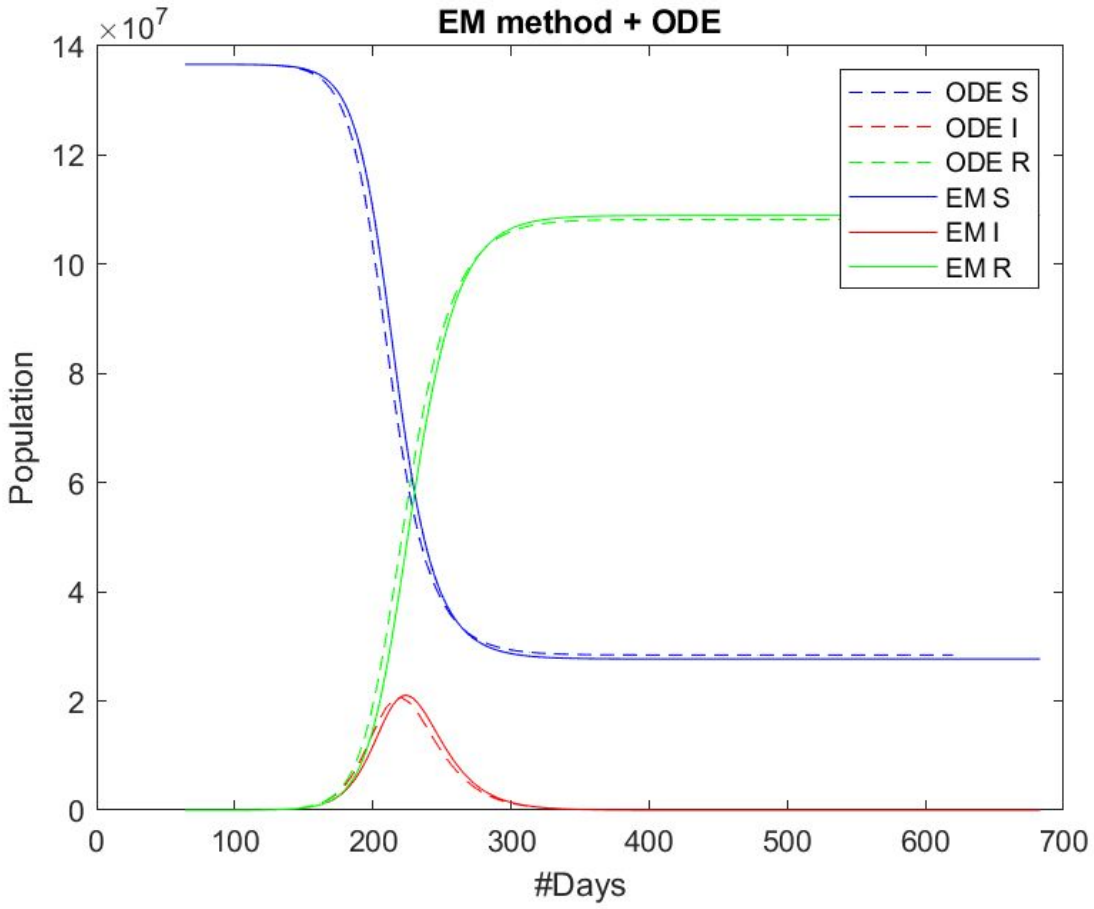
The nationwide lockdown was initially enforced for 3 weeks from 25th March 2020 onwards but in light of developing conditions was extended until 31st May 2020. In ground-reality, how strictly it was enforced and followed varies greatly from region to region but we still assume it to have some impact when we meta-analyse the parameters on the total number of cases nationally.

```
COVID-19 in India - Phase 2 Analysis - Lockdown
Total population=136641750    Initial Infectives=602    Basic Reproduction Number=1.9853
Start Date:2020-03-25    End Date:2020-05-31    Transmission Rate(Beta)=0.15798    Removal Rate(Gamma)=0.079575
Optimization status=1    Optimized error function=3296.0176
Probabilty of major outbreak=1    Total simulated number of cases=108218836.6941
Deterministic Status: Epidemic
```

Properties obtained for phase-2 analysis



Understanding accuracy of estimation and comparison of actual data, deterministic and stochastic models for phase-2 analysis



Deterministic and stochastic simulation for longer duration for phase-2 analysis

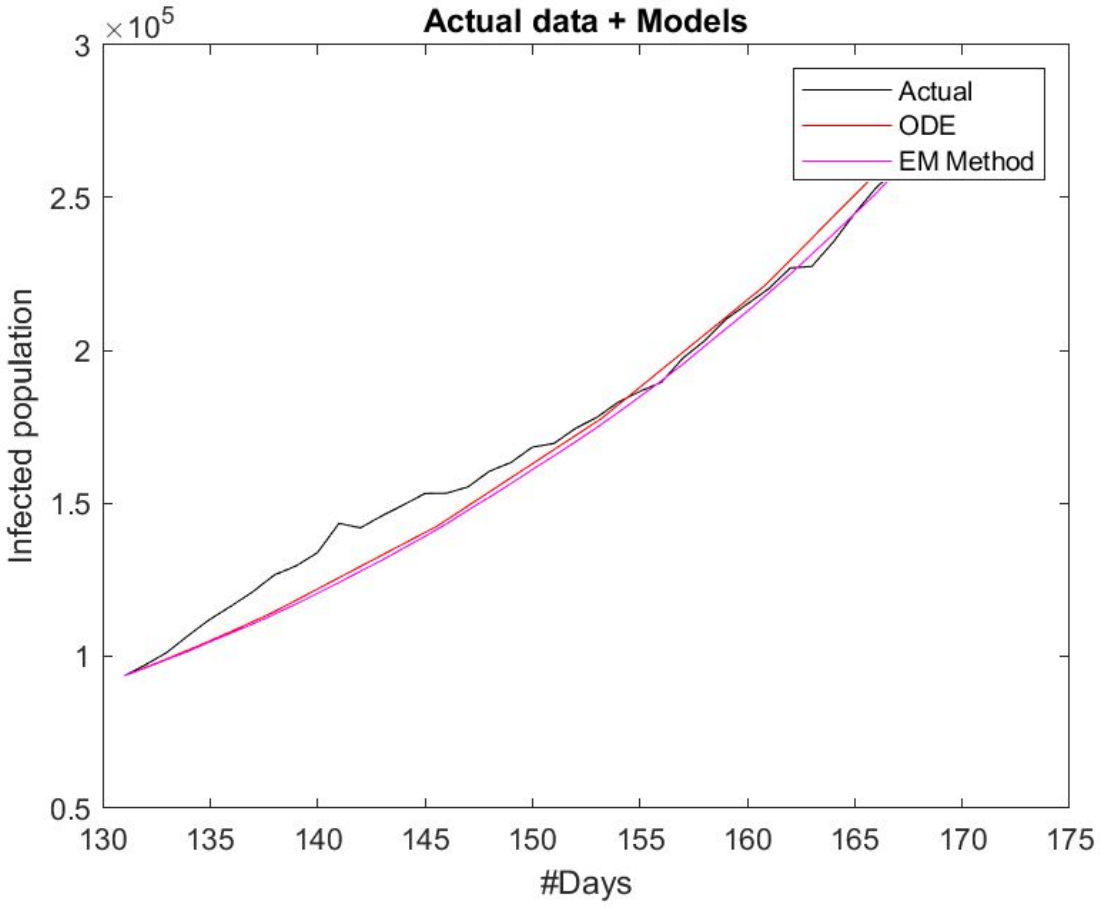
On direct comparison with the first phase we can easily observe the number of total cases is greatly reduced. Thus in an absolute sense the lockdown was beneficial in curbing the spread of the disease to some extent at least. Whether it helped the general population in a more generalized sense would require an additional parallel study of the economic losses suffered and its impacts.

5.8.3.3. Phase 3

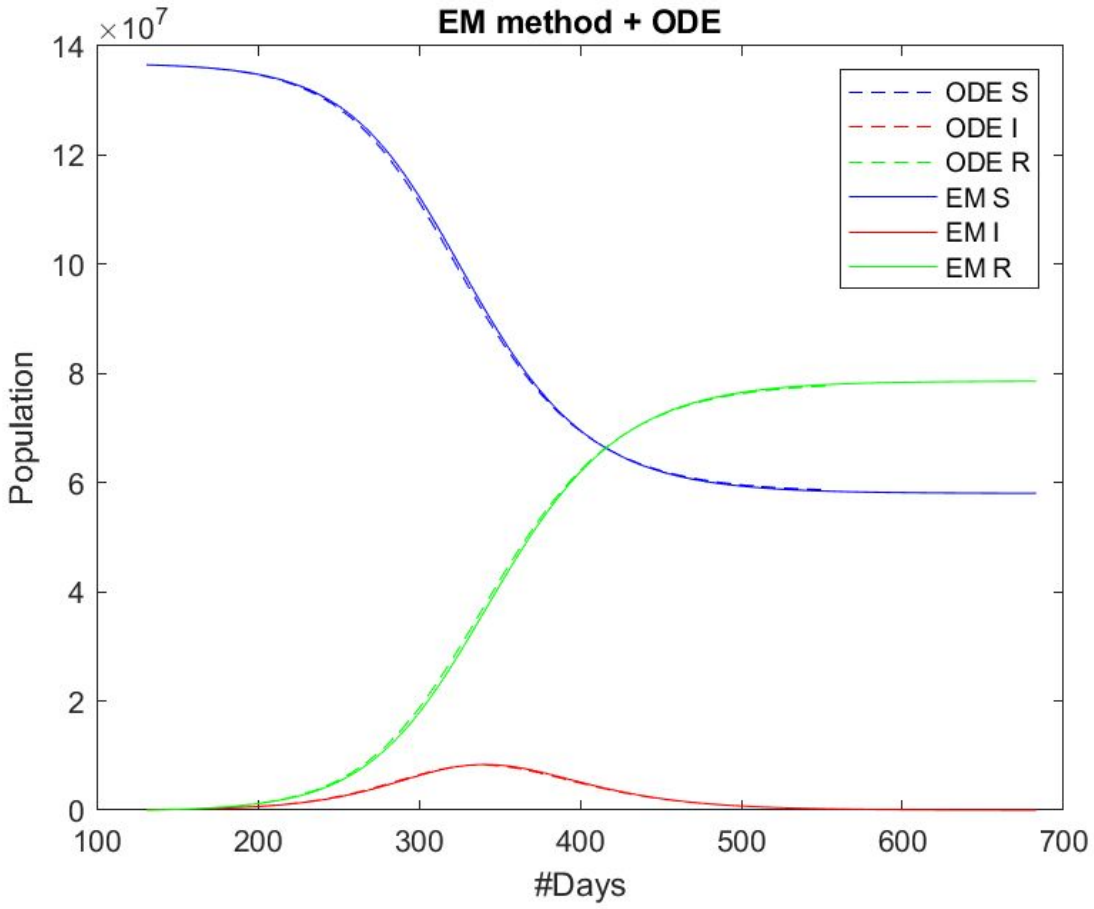
Due to such an extended lockdown period and growing unrest amongst masses suffering from loss of jobs or daily wages the government decided to end the lockdown on 31st May 2020 in an orderly fashion according to per-region requirements. From this time on, almost till present day, i.e. 10th July 2020 is the last phase taken here.

```
COVID-19 in India - Phase 3 Analysis - Post-Lockdown
Total population=136641750    Initial Infectives=93349    Basic Reproduction Number=1.4847
Start Date:2020-05-31    End Date:2020-07-10    Transmission Rate(Beta)=0.089125    Removal Rate(Gamma)=0.060028
Optimization status=1    Optimized error function=1778.8977
Probabilty of major outbreak=1    Total simulated number of cases=77715367.562
Deterministic Status: Epidemic
```

Properties obtained for phase-3 analysis



Understanding accuracy of estimation and comparison of actual data, deterministic and stochastic models for phase-3 analysis



Deterministic and stochastic simulation for longer duration for phase-3 analysis

For the last phase we observe that the total number of people estimated to catch the disease is even further reduced without a government enforced lockdown. This possibly highlights the consciousness developed in individuals to maintain social distance and take other precautionary measures.

6. Conclusion

We used the data of pneumococcus spread in children of age two years old and under and found the value of constants of SIS epidemic model using curve fitting (via scipy library of python which finds the best fit of a function in a given data) and then using those values we simulated the DTMC SIS process on MATLAB. We found that the results from the simulation fit the data well mostly but since there we are sampling paths from DTMC SIS model so the simulation may sometimes go a bit farther from the data points, but on average the simulation fits the data well since the paths are centered around deterministic SIS process whose equations were used during curve fitting process.

In our research for the SIR model, we found that the stochastic model is better in modelling the epidemic than the deterministic model as the randomness of the infections gets accounted for in the stochastic model. The data also suggests that the lockdown has been effective in curbing the covid-19 cases because the curve for the cases is considerably flatter there.

The metapopulation model is more accurate in modelling the epidemic as it is a better representation of the epidemic, however it requires a lot of computational power. In the simulation part, the Gillespie algorithm is a more detailed stochastic simulation than the Euler Maruyama method as it is slower because of it being computationally expensive.

We used the data of Susceptible, Infected and Recovered individuals and found the value of beta and gamma through the estimation methods described above. The estimation of parameters using Machine Learning as can be seen from graphs is quite accurate with the values of beta and gamma coming out to be 0.2531 and 0.1829 respectively. The values predicted reproduced the results in accordance with the data available.

In conclusion, the accuracy of these simulations demonstrates their potential in further policy formulation. They also help us understand how effective the lockdown or other measures are relative to the normal day to day function.

7. Appendix

The codes are given in the github links given below as well as they have been written in this section.

SIR codes https://github.com/prisha2610/ASP-SIR
SIS codes https://github.com/nihirag/SIS-DTMC-Model

SIS codes

Python code for curve fitting in SIS model for pneumococcus data:-

```
import numpy as np
from scipy.optimize import curve_fit
from matplotlib import pyplot as plt
x, y = [], []
for line in open('my_data.txt', 'r'):
 values = [float(s) for s in line.split()]
 x.append(values[0])
 y.append(values[1])
N=150000
I=5000
def test(x, a, b, c):
   terms=(b*N)-a-c
   return 1/((b/terms)*(1-np.exp(-terms*x))+(1/I)*np.exp(-terms*x))
param, param_cov = curve_fit(test, x, y)
print("function coefficients:")
print(param)
print("Covariance of coefficients:")
print(param_cov)
k=(param[1]*N)-param[0]-param[2]
ans = (1/((param[1]/k)*(1-np.exp(-k*x))+(1/I)*np.exp(-k*x)))
plt.plot(x, y, 'o', color ='red', label ="data")
plt.plot(x, ans, '--', color ='blue', label ="optimized data")
plt.legend()
plt.show()
```

MATLAB code for simulation of figure:-

```
clear
set(0,'DefaultAxesFontSize', 18)
beta=1;
g=0.25;
b=0.25;
R0=beta/(b+g)
N=100;
init=2;
```

```
dt=0.01;
time=25;
sim=3;
for j=1:sim
i(1)=init;
for t=1:time/dt
r=rand; % uniform random number
birth=beta*i(t)*(N-i(t))/N*dt;
death=(b+g)*i(t)*dt;
if r<=birth
i(t+1)=i(t)+1;
elseif r>birth & r<=birth+death
i(t+1)=i(t)-1;
else
i(t+1)=i(t);
end
end
if j==1 stairs([0:dt:time],i,'r-','LineWidth',2);
hold on
elseif j==2 stairs([0:dt:time],i,'g-','LineWidth',2);
else
stairs([0:dt:time],i,'b-','LineWidth',2);
end
end
y(1)=init;
for k=1:time/dt
   y(k+1)=y(k)+dt*(beta*(N-y(k))*y(k)/N-(b+g)*y(k));
end
plot([0:dt:time],y,'k--','LineWidth',2);
hold off
axis([0,25,0,80]);
xlabel('Time');
ylabel('Number of Infectives');
```

MATLAB code for SIS simulation

```
clear
set(0,'DefaultAxesFontSize', 18)
beta=0.0000002857*100000;
g=0.02011;
b=0.00137363;
N=150000;
R0=(beta)/(b+g);
```

```
init=5000;
dt=0.01;
time=5000;
i(1)=init;
for t=1:time/dt
r=rand; % uniform random number
birth=beta*i(t)*(N-i(t))/N*dt;
death=(b+g)*i(t)*dt;
if r<=birth
i(t+1)=i(t)+1;
elseif r>birth && r<=birth+death
i(t+1)=i(t)-1;
else
i(t+1)=i(t);
end
end
stairs([0:dt:time],i,'r-','LineWidth',2);
hold on
y(1)=init;
for k=1:time/dt
y(k+1)=y(k)+dt*(beta*(N-y(k))*y(k)/N-(b+g)*y(k));
end
plot([0:dt:time],y,'k--','LineWidth',2);
hold off
axis([0,5000,0,100000]);
xlabel('t')
ylabel('I(t)');
```

SIR codes

data_extraction.py

```
import re

fn1='time-series-countries.txt'
fn2='date-India-confirmed-recovered-dead.csv'
fh1=open(fn1)
fh2=open(fn2,'a')
fh2.write('Date,c,r,d\n')
for l in fh1:
    match=re.findall('(.*),India.+(,\d*,\d*,\d*)',l)
    if len(match)<1:
        continue
    a=match[0];
    l2=a[0]+a[1]
    fh2.write(l2+'\n')
```

main_Script.m

```
% Main Script to be executed
% Saves plots and displays attributes
% -----------------------------------
% The variables used are -
% N -- Total population
% days -- Total number of days for which data is available
% x -- Start index of time period
% y -- End index of time period
% i -- Initial number of infectives
% beta -- Estimated value of transmission parameter
% gamma -- Estimated value of removal parameter
% flag -- Epidemic status according to Deterministic Theorem
% p -- Probability of a major outbreak
% f_spread -- Total cases according to simulation
% f -- Optimization status
% fminval -- Optimization function value
% f1 -- Figure for showing actual data + ODE
% f2 -- Figure for showing ODE over longer period of time
% f3 -- Figure for showing EM-plot over longer period of time
% -----------------------------------
% The functions used are -
% det_ode -- ODE plot
% isEpidemic -- Epidemic status according to Deterministic Theorem
% P_majOutBr -- Probability of a major outbreak
% EM_method -- EM Method simulation
% gillespie -- Gillespie Algorithm simulation
% get_data -- Setting up actual data
% view_data -- Plotting actual data
% parest -- Estimation of beta and gamma
% -----------------------------------

global days beta gamma
global x y i
global I N date
global f1 f2 f3

close all
beta=0.08;
gamma=0.03;
N=1000;
i=10;
x=1;
days=200;
y=days*4;
f_spread=det_ode;
```

```
flag=isEpidemic;
p=P_majOutBr;
disp("Theoretical");
disp("Total Population="+N+"    Initial Infectives="+i...
    +"    Basic Reproduction Number="+(beta/gamma));
disp("Start Date:"+x+"    End Date:"+y+...
    "    Transmission Rate(Beta)="+beta+"    Removal Rate(Gamma)="+gamma)
disp("Probability of major outbreak="+p+"    Total simulated number of
cases="+f_spread);
if flag
    disp("Deterministic Status: Epidemic");
else
    disp("Deterministic Status: Disease-free equilibrium");
end
disp(" ");
EM_method;
gillespie;
saveas(f2,'ODE_theory.png');
saveas(f3,'EM_theory.png')
saveas(f1,'Gillespie_theory.png')

get_data;

days=length(I);

close all
x=43;
y=64;
i=I(x);
view_data;
[f,fminval]=parest;
f_spread=det_ode;
flag=isEpidemic;
p=P_majOutBr;
disp("COVID-19 in India - Phase 1 Analysis - Pre-Lockdown")
disp("Total population="+N+"    Initial Infectives="+i...
    +"    Basic Reproduction Number="+(beta/gamma));
disp("Start Date:"+date(x)+"    End Date:"+date(y)+...
    "    Transmission Rate(Beta)="+beta+"    Removal Rate(Gamma)="+gamma);
disp("Optimization status="+f+"    Optimized error function="+fminval);
disp("Probability of major outbreak="+p+"    Total simulated number of
cases="+f_spread);
if flag
    disp("Deterministic Status: Epidemic");
else
    disp("Deterministic Status: Disease-free equilibrium");
end
disp(" ");
```

```
gillespie;
EM_method;
saveas(f1,'Estimation_1.png');
saveas(f2,'ODE_1.png');
saveas(f3,'EM_1.png');

close all
x=64;
y=131;
i=I(x);
view_data;
[f,fminval]=parest;
f_spread=det_ode;
flag=isEpidemic;
p=P_majOutBr;
disp("COVID-19 in India - Phase 2 Analysis - Lockdown")
disp("Total population="+N+"    Initial Infectives="+i...
    +"    Basic Reproduction Number="+(beta/gamma));
disp("Start Date:"+date(x)+"    End Date:"+date(y)+...
    "    Transmission Rate(Beta)="+beta+"    Removal Rate(Gamma)="+gamma);
disp("Optimization status="+f+"    Optimized error function="+fminval);
disp("Probability of major outbreak="+p+"    Total simulated number of
cases="+f_spread);
if flag
    disp("Deterministic Status: Epidemic");
else
    disp("Deterministic Status: Disease-free equilibrium");
end
disp(" ");
%gillespie;
EM_method;
saveas(f1,'Estimation_2.png');
saveas(f2,'ODE_2.png');
saveas(f3,'EM_2.png');

close all
x=131;
y=days;
i=I(x);
view_data;
[f,fminval]=parest;
f_spread=det_ode;
flag=isEpidemic;
p=P_majOutBr;
disp("COVID-19 in India - Phase 3 Analysis - Post-Lockdown")
disp("Total population="+N+"    Initial Infectives="+i...
    +"    Basic Reproduction Number="+(beta/gamma));
disp("Start Date:"+date(x)+"    End Date:"+date(y)+...
```

```
    "    Transmission Rate(Beta)="+beta+"    Removal Rate(Gamma)="+gamma);
disp("Optimization status="+f+"    Optimized error function="+fminval);
disp("Probability of major outbreak="+p+"    Total simulated number of
cases="+f_spread);
if flag
    disp("Deterministic Status: Epidemic");
else
    disp("Deterministic Status: Disease-free equilibrium");
end
disp(" ");
%gillespie;
EM_method;
saveas(f1,'Estimation_3.png');
saveas(f2,'ODE_3.png');
saveas(f3,'EM_3.png');

close all
x=43;
y=days;
i=I(x);
view_data;
[f,fminval]=parest;
f_spread=det_ode;
flag=isEpidemic;
p=P_majOutBr;
disp("COVID-19 in India - Overall Analysis");
disp("Total population="+N+"    Initial Infectives="+i...
    +"    Basic Reproduction Number="+(beta/gamma));
disp("Start Date:"+date(x)+"    End Date:"+date(y)+...
    "    Transmission Rate(Beta)="+beta+"    Removal Rate(Gamma)="+gamma);
disp("Optimization status="+f+"    Optimized error function="+fminval);
disp("Probability of major outbreak="+p+"    Total simulated number of
cases="+f_spread);
if flag
    disp("Deterministic Status: Epidemic");
else
    disp("Deterministic Status: Disease-free equilibrium");
end
%gillespie;
EM_method;
saveas(f1,'Estimation_overall.png');
saveas(f2,'ODE_overall.png');
saveas(f3,'EM_overall.png');
```

---

get_data.m

```
% Setting up the data
% ------------------------------------
```

```
% The variables used are -
% S -- Susceptible population
% I -- Infected population
% R -- Removed population
% -----------------------------------

function []=get_data()

    global S I R date
    global N

    T=readtable('date-India-confirmed-recovered-dead.csv');
    N=136641750;

    date=string(T.Date);
    R=T.r+T.d;
    I=T.c-T.r-T.d;
    S=N-T.c;

end
```

---

view_data.m

```
% Plotting Actual Data
% -----------------------------------

function []= view_data()

    global x y I
    global f1

    f1=figure;
    plot(x:y,I(x:y),'-k');
    hold on;
    legend('Actual');
    title('Actual data + Models');
    xlabel('#Days');
    ylabel('Infected population');

end
```

---

parest.m

```
% Estimating parameters
% -----------------------------------
% The variables are -
% maxiters -- Maximum number of iteration for the fminsearch function
% b0 -- Array containing the initial values of beta and gamma
```

```
% options -- Setting attributes for fminsearch
% b -- Values of the parameters with minimum loss
% fminval -- Minimum value of error found
% f -- Status of fminsearch:
%         1 if minimum is reached, 0 if minimum not reached, -1 if error
function not convergent
% ------------------------------------
% The functions used are-
% iniGuess -- Function for initial guess of parameters
% fminsearch -- An optimizer provided by MATLAB for minimizing the value of
optim_fun
% optim_fun -- The function to optimize
% ------------------------------------

function [f,fminval] = parest()

    global beta gamma

    b0 = iniGuess();

    maxiters = 1000;

    options = optimset('Display','off','MaxIter',maxiters,...

'MaxFunEvals',maxiters,'TolFun',1e-6,'TolX',1e-6,'PlotFcn',@optimplotfval);

    [b, fminval,f] = fminsearch(@optim_fun, b0, options);
    warning('on')

    beta=b(1);
    gamma=b(2);

end
```

---

iniGuess.m

```
% Initial guess for parameters
% ------------------------------------
% The variables used are -
% b0 -- initial guess for beta and gamma
% ------------------------------------
% The functions used are -
% rand -- Uniformly random generator ~ U(0,1)
% ------------------------------------

function b0 = iniGuess()

b0(1) = rand;           %random value of beta
```

```
b0(2) = rand;           %random value of gamma

end
```

---

omptim_fun.m

```
% Function used for optimization and finding the solution for beta and gamma
% ------------------------------------
% The variables are -
% m -- Time-span
% sol -- Solution of ODE
% f -- Mean squared error
% ------------------------------------
% The functions used are-
% ode45 -- Non-stiff ODE solver
% odefunc -- Sets SIR Model
% norm -- Normalizing function
% ------------------------------------

function f = optim_fun(params)

    global S I x y

    m=y-x+1;

    %solve ODE
    try
        warning('off')
    [tsol,sol] = ode45(@(t,y) odefunc(t,y,params(1),params(2)),0:m-1,
[S(x),I(x)]);
        warning('on')
    catch
        f=NaN;
        warning('on')
        return
    end

    %calculate optimization function
    f =  (norm((S(x:y) - sol(:,1))) + norm((I(x:y) - sol(:,2))))/m;

end
```

---

odefunc.m

```
% Setting the SIR model
% ------------------------------------
% The variables used are -
% t -- Time-steps
```

```
% y -- [S;I]
% dy -- [dS;dI]
% -----------------------------------

function dy = odefunc(t,var,beta,gamma)

global N

dy=zeros(2,1);

dy(1)=-(beta/N)*var(1)*var(2);                %dS=-(beta/N)*s*i
dy(2)=(beta/N)*var(1)*var(2)-gamma*var(2);    %dI=(beta/N)*s*i-gamma*i

end
```

---

det_ode.m

```
% Deterministic curve plot
% -----------------------------------
% The variables used are -
% t --
% var --
% id --
% f_spread -- final number of cases
% -----------------------------------
% The functions used are-
% ode45 -- Non-stiff ODE solver
% odefunc -- Sets SIR Model
% -----------------------------------

function [f_spread]=det_ode()

    warning ('off');

    global N days i
    global beta gamma x y
    global f1 f2 f3

    %obtaining ODE solution
    [t,var] = ode45(@(t,var) odefunc(t,var,beta,gamma), [x 4*days-x], [N-i i]);

    warning ('on');

    f_spread=N-var(end,1)-var(end,2);

    if x==1 && y==800
        f1=figure;
        plot(t,var(:,1),'--b');
```

```
        hold on;
        plot(t,var(:,2),'--r');
        plot(t,N-var(:,1)-var(:,2),'--g');
        legend('ODE S','ODE I','ODE R');
        title('Gillespie Algo + ODE');
        xlabel('#Days');
        ylabel('Population');

        f2=figure;
        plot(t,var(:,1),'-b');
        hold on;
        plot(t,var(:,2),'-r');
        plot(t,N-var(:,1)-var(:,2),'-g');
        legend('Susceptible','Infected','Removed');
        title('ODE');
        xlabel('#Days');
        ylabel('Population');
    else
        id=find(t<=y);
        figure(f1);
        plot(t(id),var(id,2),'-r','DisplayName','ODE');
    end


    f3=figure;
    plot(t,var(:,1),'--b');
    hold on;
    plot(t,var(:,2),'--r');
    plot(t,N-var(:,1)-var(:,2),'--g');
    legend('ODE S','ODE I','ODE R');
    title('EM method + ODE');
    xlabel('#Days');
    ylabel('Population');

end
```

---

isEpidemic.m

```
% Deterministic Epidemic Theorem
% -----------------------------------
% The variables used are -
% flag -- Indicates epidemic status
%         0 if disease-free equilibrium and 1 if epidemic
% R0 -- basic reproduction number
% s -- Initial number of infectives
% -----------------------------------

function flag = isEpidemic()
```

```
    global N beta gamma i

    flag=0;
    R0=beta/gamma;
    s=N-i;

    if s*R0/N>1    %condition obtained from theorem
        flag=1;
    end

end
```

P_majOutBr.m

```
% Deterministic Epidemic Theorem
% -----------------------------------
% The variables used are -
% p -- Probability of major outbreak
% R0 -- basic reproduction number
% -----------------------------------

function p = P_majOutBr()

    global beta gamma i
    p=0;
    R0=beta/gamma;

    if R0>1             %derivation of condition from branching process
assumption
        p=1-(1/R0)^i;
    end

end
```

---

gillespie.m

```
% Gillespie Algorithm
% -----------------------------------
% The variables used are -
% t -- Time-steps
% s -- Susceptible individuals at particular time step
% I -- Infected individuals at each time step
% u -- Vector of 2 uniformly random numbers to simulate intervent time and
event occurrence
% lambda -- total rate of either event occurring
% -----------------------------------
% The functions used are -
```

```
% rand -- Uniformly random generator ~ U(0,1)
% ------------------------------------

function [] = gillespie()

    global N beta gamma
    global i x y
    global f1

    t=x;                                        %initial time
    S=N-i;                                      %initial susceptible population
    I=i;                                        %initial susceptible population

    while(t(1)<y && I(1)~=0)                    %time for occurence of event can't
exceed total time
        u=rand(2,1);
        lambda=(beta*S(1)*I(1))/N+gamma*I(1);
        t =[t(1)-log(u(1))/lambda; t];          %simulation of time at which next
event occurs
        if u(2)<=beta*S*I(1)/(N*lambda)         %condition for transmission to
occur
            if S(1)~=0
                S=[S(1)-1; S];
                I=[I(1)+1; I];
            else
                S=[S(1); S];
                I=[I(1); I];
            end
        else                                    %condition for removal to occur
            S=[S(1); S];
            I=[I(1)-1; I];
        end
    end
    t=[y; t];
    S=[S(1); S];
    I=[I(1); I];

    figure(f1);

    if x==1 && y==800
        plot(t,S,'-b','DisplayName','Gillespie S');
        hold on;
        plot(t,I,'-r','DisplayName','Gillespie I');
        plot(t,N-S-I,'-g','DisplayName','Gillespie R');
        xlabel('#Days');
        ylabel('Population');
    else
        plot(t,I,'-y','DisplayName','Gillespie Algo');
```

```
    end

end
```

EM_method.m

```
% EM simulation
% ----------------------------------
% The variables used are -
% dt -- Fractional time-step
% steps -- Total #time-steps
% S -- Simulated number of susceptibles
% I -- Simulated number of infectives
% R -- Simulated number of removed
% t -- Particular time instant
% eta -- vector of two normally random numbers to simulate Wiener processes
% f -- Expected transmitted and removed populations at t
% gw -- Covariance values for time-step t (simplified vector form)
% s -- Simulated susceptible population for time-step t+1
% i1 -- Simulated infected population for time-step t+1
% ----------------------------------
% The functions used are -
% randn -- Standard normally random number generator ~ N(0,1)
% ----------------------------------

function []=EM_method()

    global N days beta gamma
    global i x y
    global f1 f3

    dt=1;
    steps=(4*days-x)*dt;
    S=zeros(steps,1);
    I=zeros(steps,1);

    S(1)=N-i;                       %initial number of susceptibles
    I(1)=i;                         %initial number of infectives

    for t=1:steps-1
        eta=randn(2,1);
        f(1)=beta*S(t)*I(t)*dt/N;
        f(2)=gamma*I(t)*dt;
        gw=sqrt(f).*eta;
        s=S(t)-f(1)-gw(1);
        i1=I(t)+f(1)-f(2)+gw(1)-gw(2);
         if i1<=0                   %end of epidemic condition
             S(t+1:steps)=s+i1;
```

```
            I(t+1:steps)=0;
            break;
        elseif s<0                  %non-negativity of susceptible population
condition
            S(t+1)=0;
            I(t+1)=i1-s;
         else
            S(t+1)=s;
            I(t+1)=i1;
         end
    end
    R=N-S-I;                        %S+I+R=N at any time-step

    if days~=200
        figure(f1);
        plot(x:y-1,I(1:y-x),'-m','DisplayName','EM Method');
    end

    figure(f3);
    plot(x:steps+x-1,S,'-b','DisplayName','EM S');
    hold on;
    plot(x:steps+x-1,I,'-r','DisplayName','EM I');
    plot(x:steps+x-1,R,'-g','DisplayName','EM R');

end
```

8. References

1. *A primer on stochastic epidemic models: Formulation, numerical simulation, and analysis* https://doi.org/10.1016/j.idm.2017.03.001
2. *Stochastic Epidemic Models and their Statistical Analysis* https://link.springer.com/book/10.1007/978-1-4612-1158-7
3. *A spatial model of CoVID-19 transmission in England and Wales: early spread and peak timing* https://www.medrxiv.org/content/10.1101/2020.02.12.20022566v1
4. *Stochastic spatial model for Coronavirus spread in UK (metapopulation plus SEIR framework)* https://www.youtube.com/watch?v=GRuPAgR-Guc&t
5. *Methods for studying stochastic disease dynamics* *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2705976/*
6. *COVID-19 dataset* https://github.com/datasets/covid-19/blob/master/data/countries-aggregated.csv
7. *Population of India* https://data.worldbank.org/indicator/SP.POP.TOTL?locations=IN
8. *Data license for COVID-19 data* https://opendatacommons.org/licenses/pddl/1-0/
9. *Info related to pneumococcus* https://www.hps.scot.nhs.uk/a-to-z-of-topics/pneumococcal-disease/
10. https://www.sciencedirect.com/science/article/pii/S0377042710001676