SEIRD-LQ
Population Modeling of Multiple Contagious Viruses

Ayan Ahmad (2K19/EP/027)

Dhruv Tyagi (2K19/EP/032)

Supervisor: **Dr. Ajeet Kumar**

Engineering Physics, Sem III

Department of Applied Physics

Delhi Technological University (formally DCE)

November, 2020

Written in LaTeX

Contains hyperlinked table of contents, references in blue & MATLAB editor listing.

Acknowledgements

This project was a part of our mid-semester evaluation being done under the supervision of Dr. Ajeet Kumar who is a Professor in the Department of Applied Physics at Delhi Technological University. The research interests include Fiber and Integrated Optics, Numerical Modelling, Nonlinear Optics.

We spent a total of about 13 weeks working on this project, from August 2020 to November 2020. We would first like to thank Dr. Ajeet Kumar for giving us the opportunity to not only work on the project from a research and report point of view but also presenting the work we have been doing for the past few weeks. This opportunity has been a great learning experience for us, from the topics we chose, to reading research papers, trying to come up with new modifications, to the project writing and presentation skills too. It under his guidance that we were able to learn so much in such a short span of time about MATLAB and concepts of numerical approximation in modeling.

Abstract

We build a population model that quantitatively describes the expected trend in the spread of a contagious virus on a macroscopic level. The model calculates the rate of contagion with three central components that contribute to the spread, first of which is population category which can be classified into four types - susceptible, infectious, and removed (immune or deceased). These population types, combined with the possibility of catching the virus twice, results in models that are often abbreviated to be known as the SEIRS model. The second component is the intrinsic nature and properties of the virus such as its deadliness and symptom-free growth duration. The third central component is mesoscopic factors such as the percentage of people following social distancing and other precautions, which decides the rate of transmission or fluid contagiousness. We wanted to do further than that. So we decided to take this model further, forming a model we call the *SEIRD-LQ* model, which will be explained in further sections.

Contents

1

Introduction

1.1 | Motivation

With the ongoing pandemic and an interest in mathematical modeling of systems, the SIR model is what first came to our minds. On further research done as a part of our literature review, we found that there models that predict not only more trends of contagious viruses but also does it more accurately. For this reason we moved on to the SEIR model. As a part of our literature review of contagiousness of viruses, we found out that there are many cases that are not included in the modeling of viruses. For this reason, we first went on to formulate a modified version of SEIR model.

1.2 | Aims and Objectives

We will have mainly two aims as a part of our project, the first being to develop the aforementioned modified version of SEIR model, namely the *SEIRD-LQ* model. The abbriviation stands for the following-

- S $\rightarrow$ Susceptible
- E $\rightarrow$ Exposed
- I $\rightarrow$ Infected
- R $\rightarrow$ Recovered
- D $\rightarrow$ Dead
- L $\rightarrow$ Lockdown
- Q $\rightarrow$ Quarantine

The second aim for the project is to form an interactive simulation for the model on MATLAB which gives a better visual trend display to the user under different circumstances.

2

Background & Literature Overview

2.1 | Compartmental Population Models in Epidemiology

A population model is a type of mathematical model that is applied to the study of population dynamics. Models allow a better understanding of how complex interactions and processes work. Modeling of dynamic interactions in nature can provide a manageable way of understanding how numbers change over time or in relation to each other. Many patterns can be noticed by using population modeling as a tool. Worster (1994)

Ecological population modeling is concerned with the changes in parameters such as population size and age distribution within a population. This might be due to interactions with the environment, individuals of their own species, or other species.Singh and Uyenoyama (2004)

Population models are used to determine maximum harvest for agriculturists, to understand the dynamics of biological invasions, and for environmental conservation. Population models are also used to understand the spread of parasites, viruses, and disease (Singh and Uyenoyama (2004)). Another way populations models are useful are when species become endangered. Population models can track the fragile species and work and curb the decline.

2.1.1 | SIR Model

The SIR model (Harko et al. (2014) and Beckley et al. (2013)) is one of the simplest compartmental models, and many models are derivatives of this basic form. The model consists of three compartments:

- **S** $\rightarrow$ The number of susceptible individuals. When a susceptible and an infectious individual come into "infectious contact", the susceptible individual contracts the disease and transitions to the infectious compartment.
- **I** $\rightarrow$ The number of infectious individuals. These are individuals who have been infected and are capable of infecting susceptible individuals.

- **R** $\rightarrow$ for the number of removed (and immune) or deceased individuals. These are individuals who have been infected and have either recovered from the disease and entered the removed compartment, or died. It is assumed that the number of deaths is negligible with respect to the total population. This compartment may also be called "recovered" or "resistant".

This model is reasonably predictive (Yang et al. (2020)) for infectious diseases that are transmitted from human to human, and where recovery confers lasting resistance, such as measles, mumps and rubella.

Spatial SIR model simulation. Each cell can infect its eight immediate neighbors. These variables (S, I, and R) represent the number of people in each compartment at a particular time. To represent that the number of susceptible, infectious and removed individuals may vary over time (even if the total population size remains constant), we make the precise numbers a function of t (time): $S(t)$, $I(t)$ and $R(t)$. For a specific disease in a specific population, these functions may be worked out in order to predict possible outbreaks and bring them under control. (Yang et al. (2020))

As implied by the variable function of t, the model is dynamic in that the numbers in each compartment may fluctuate over time. The importance of this dynamic aspect is most obvious in an endemic disease with a short infectious period, such as measles in the UK prior to the introduction of a vaccine in 1968. Such diseases tend to occur in cycles of outbreaks due to the variation in number of susceptibles ($S(t)$) over time. During an epidemic, the number of susceptible individuals falls rapidly as more of them are infected and thus enter the infectious and removed compartments. The disease cannot break out again until the number of susceptibles has built back up, e.g. as a result of offspring being born into the susceptible compartment.

Figure 2.1: States in an SIR epidemic model and the rates at which individuals transition between them

2.1.1.1 | Transmission Rates

For the full specification of the model, the arrows should be labeled with the transition rates between compartments. Between S and I, the transition rate is assumed to be $\frac{d(S/N)}{dt} = -\beta \frac{SI}{N_2}$, where N is the total population, β is the average number of contacts per person per time, multiplied by the probability of disease transmission in a contact between a susceptible and an infec-

tious subject, and SI/N_2 is the fraction of those contacts between an infectious and susceptible individual which result in the susceptible person becoming infected. (This is mathematically similar to the law of mass action in chemistry in which random collisions between molecules result in a chemical reaction and the fractional rate is proportional to the concentration of the two reactants).

Between I and R, the transition rate is assumed to be proportional to the number of infectious individuals which is γI. This is equivalent to assuming that the probability of an infectious individual recovering in any time interval dt is simply γdt. If an individual is infectious for an average time period D, then $\gamma = \frac{1}{D}$. This is also equivalent to the assumption that the length of time spent by an individual in the infectious state is a random variable with an exponential distribution. The "classical" SIR model may be modified by using more complex and realistic distributions for the I-R transition rate (e.g. the Erlang distribution - Krylova and Earn (2013)).

For the special case in which there is no removal from the infectious compartment ($\gamma = 0$), the SIR model reduces to a very simple SI model, which has a logistic solution, in which every individual eventually becomes infected.

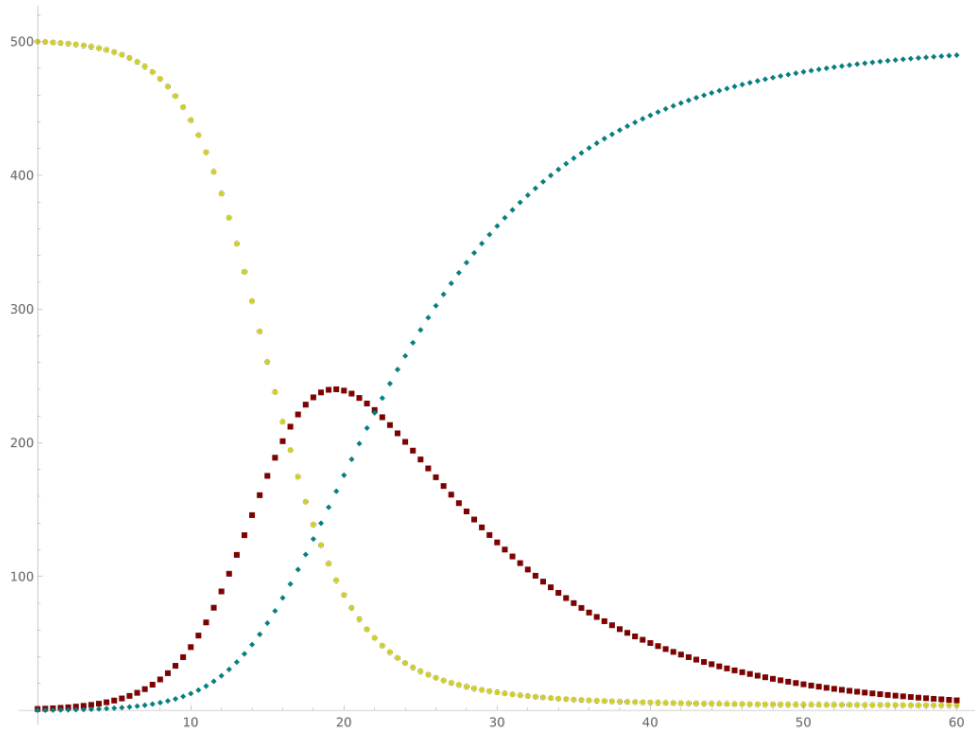

Figure 2.2: Yellow=Susceptible, Maroon=Infectious, Teal=Recovered

2.1.1.2 | Mathematical Modeling

The dynamics of an epidemic, for example, the flu, are often much faster than the dynamics of birth and death, therefore, birth and death are often omitted in simple compartmental models. The SIR system without so-called vital dynamics (birth and death, sometimes called demography) described above can be expressed by the following set of ordinary differential equations: (Beckley et al. (2013) and Hethcote (2000))

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta IS}{N}\\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I\\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

where S is the stock of susceptible population, I is the stock of infected, R is the stock of removed population (either by death or recovery), and N is the sum of these three. This model was for the first time proposed by William Ogilvy Kermack and Anderson Gray McKendrick as a special case of what we now call Kermack-McKendrick theory, and followed work McKendrick had done with Ronald Ross. This system is non-linear, however it is possible to derive its analytic solution in implicit form. (Harko et al. (2014)) Firstly note that from:

$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0$$

it follows that:

$$S(t) + I(t) + R(t) = \text{ constant } = N$$

expressing in mathematical terms the constancy of population N. Note that the above relationship implies that one need only study the equation for two of the three variables. Secondly, we note that the dynamics of the infectious class depends on the following ratio:

$$R_0 = \frac{\beta}{\gamma}$$

the so-called basic reproduction number (also called basic reproduction ratio). This ratio is derived as the expected number of new infections (these new infections are sometimes called secondary infections) from a single infection in a population where all subjects are susceptible. (Bailey et al. (1975)) This idea can probably be more readily seen if we say that the typical time between contacts is $T_c = \beta^{-1}$, and the typical time until removal is $T_r = \gamma^{-1}$. From here it follows that, on average, the number of contacts by an infectious individual with others before the infectious has been removed is: T_r/T_c. By dividing the first differential equation by the third, separating the variables and integrating we get

$$S(t) = S(0)e^{-R_0(R(t)-R(0))/N}$$

where $S(0)$ and $R(0)$ are the initial numbers of, respectively, susceptible and removed subjects. Writing $s_0 = S(0)/N$ for the initial proportion of susceptible individuals, and $s_\infty = S(\infty)/N$ and $r_\infty = R(\infty)/N$ for the proportion of susceptible and removed individuals respectively in the limit $t \to \infty$, one has

$$s_\infty = 1 - r_\infty = s_0 e^{-R_0(r_\infty - r_0)}$$

(note that the infectious compartment empties in this limit). This transcendental equation has a solution in terms of the Lambert W function, [9] namely

$$s_\infty = 1 - r_\infty = -R_0^{-1} W\left(-s_0 R_0 e^{-R_0(1-r_0)}\right)$$

This shows that at the end of an epidemic that conforms to the simple assumptions of the SIR model, unless $s_0 = 0$, not all individuals of the population have been removed, so some must remain susceptible. Adriving force leading to the end of an epidemic is a decline in the number of infectious individuals. The epidemic does not typically end because of a complete lack of susceptible individuals. The role of both the basic reproduction number and the initial susceptibility are extremely important. In fact, upon rewriting the equation for infectious individuals as follows:

$$\frac{dI}{dt} = \left(R_0 \frac{S}{N} - 1\right) \gamma I$$

it yields that if:

$$R_0 \cdot S(0) > N$$

then:

$$\frac{dI}{dt}(0) > 0$$

i.e., there will be a proper epidemic outbreak with an increase of the number of the infectious (which can reach a considerable fraction of the population). On the contrary, if

$$R_0 \cdot S(0) < N$$

then

$$\frac{dI}{dt}(0) < 0$$

i.e., independently from the initial size of the susceptible population the disease can never cause a proper epidemic outbreak. As a consequence, it is clear that both the basic reproduction number and the initial susceptibility are extremely important.

2.1.1.3 | The force of infection

Note that in the above model the function:

$$F = \beta I$$

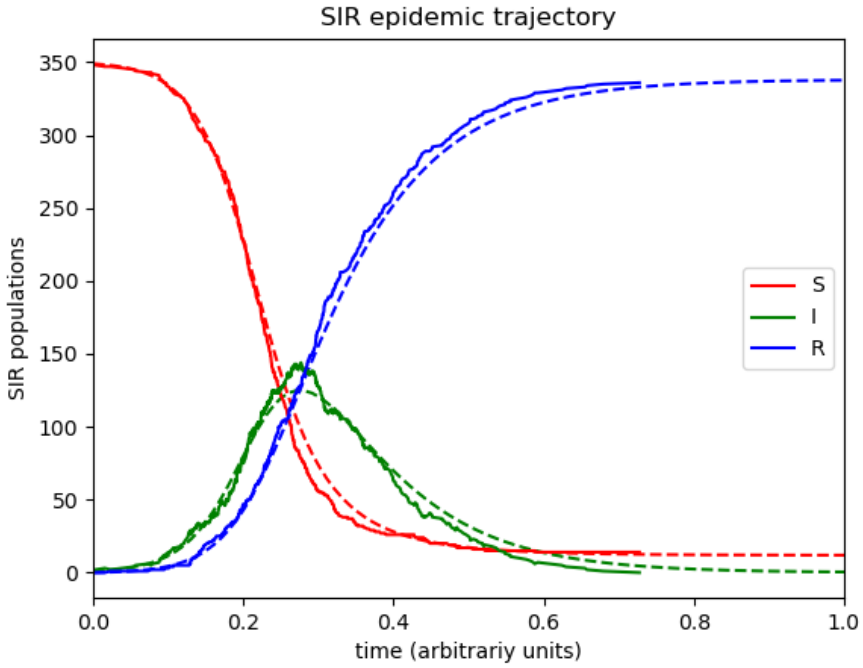

Figure 2.3: A single realization of the SIR epidemic as produced with an implementation of the Gillespie algorithm and the numerical solution of the ordinary differential equation system (dashed).

models the transition rate from the compartment of susceptible individuals to the compartment of infectious individuals, so that it is called the force of infection. However, for large classes of communicable diseases it is more realistic to consider a force of infection that does not depend on the absolute number of infectious subjects, but on their fraction (with respect to the total constant population N):

$$F = \beta \frac{I}{N}$$

Capasso [10] and, afterwards, other authors have proposed nonlinear forces of infection to model more realistically the contagion process.

2.1.1.4 | Exact analytical solutions to the SIR model

In 2014, Harko and coauthors derived an exact analytical solution to the SIR model. (Harko et al. (2014)) In the case without vital dynamics setup, for $\mathcal{S}(u) = S(t)$, ett, it corresponds to the following time parametization

$$\begin{aligned}
\mathcal{S}(u) &= S(0)u \\
\mathcal{I}(u) &= N - \mathcal{R}(u) - \mathcal{S}(u) \\
\mathcal{R}(u) &= R(0) - \rho \ln(u)
\end{aligned}$$

for

$$t = \frac{N}{\beta}\int_u^1 \frac{du^*}{u^*\mathcal{I}(u^*)}, \quad \rho = \frac{\gamma N}{\beta}$$

with intial condtitions $(\mathcal{S}(1), \mathcal{I}(1), \mathcal{R}(1)) = (S(0), N - R(0) - S(0), R(0)), \quad u_T < u < 1$ where u_T satisfies $\mathcal{I}(u_T) = 0$. By the transcendental equation for R_∞ above, it follows that $u_T = e^{-(R_\infty - R(0))/\rho}\,(= S_\infty/S(0),$ if $S(0) \neq 0)$ and $I_\infty = 0$. An equivalent analytical solution found by Miller [111112 yields

$$\begin{aligned} S(t) &= S(0)e^{-\xi(t)} \\ I(t) &= N - S(t) - R(t) \\ R(t) &= R(0) + \rho\xi(t) \\ \xi(t) &= \tfrac{\beta}{N}\int_0^t I(t^*)\,dt^* \end{aligned}$$

Here $\xi(t)$ can be interpreted as the expected number of transmissions an individual has received by time t. The two solutions are eelated by $e^{-\xi(t)} = u$. Effectively the same result can be found in the original work by Kermack and McKendrick. (Kermack and McKendrick (1927)) These solutions may be easily understood by noting that all of the terms on the right-hand sides of the original differential equations are proportional to I. The equations may thus be divided through by I, and the time rescaled so that the differential operator on the left-hand side becomes simply $d/d\tau$, where $d\tau = Idt$, i.e. $\tau = \int$ Idt. The differential equations are now all Inear, and the third equation, of the form $dR/d\tau =$ const, shows that τ and R (and ξ above) are simply linearly related.

2.1.2 | Dynamic SIR Model

Consider a population characterized by a death rate μ and birth rate Λ, and where a communicable disease is spreading. [3] The model with mass-action transmission is: $\frac{dS}{dt} = \Lambda - \mu S - \frac{\beta IS}{N}$ $\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I - \mu I$ $\frac{dR}{dt} = \gamma I - \mu R$ for which the disease-free equilibrium (DFE) is:

$$(S(t), I(t), R(t)) = \left(\frac{\Lambda}{\mu}, 0, 0\right)$$

In this case, we can derive a basic reproduction number:

$$R_0 = \frac{\beta\Lambda}{\mu(\mu + \gamma)}$$

which has threshold properties. In fact, independently from biologically meaningful initial values, one can show that.

$$\begin{aligned} &R_0 \leq 1 \Rightarrow \lim_{t\to\infty}(S(t), I(t), R(t)) = \text{DFE} = \left(\tfrac{\Lambda}{\mu}, 0, 0\right) \\ &R_0 > 1, I(0) > 0 \Rightarrow \lim_{t\to\infty}(S(t), I(t), R(t)) = \text{EE} = \left(\tfrac{\gamma+\mu}{\beta}, \tfrac{\mu}{\beta}(R_0 - 1), \tfrac{\gamma}{\beta}(R_0 - 1)\right) \end{aligned}$$

The point EE is called the Endemic Equilibrium (the disease is not totally eradicated and remains in the population). With heuristic arguments, one may show that R_0 may be read as the

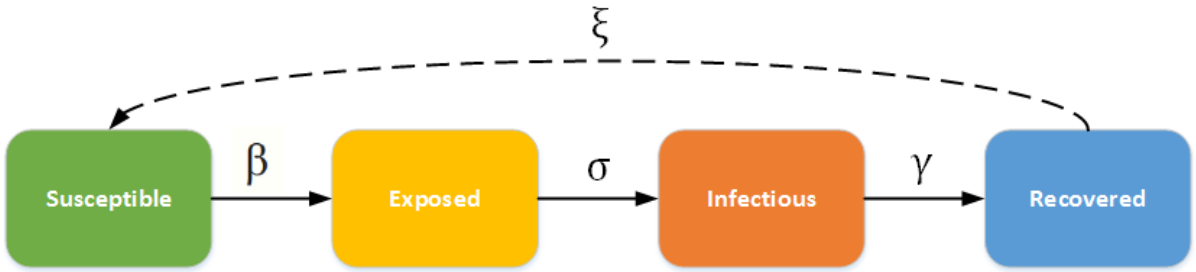

Figure 2.4: Compartmental view of SEIR model

average number of infections caused by a single infectious subject in a wholly susceptible population, the above relationship biologically means that if this number is less than or equal to one the disease goes extinct, whereas if this number is greater than one the disease will remain permanently endemic in the population.

2.1.2.1 | Steady State Solution

The expected duration of susceptibility will be $\mathrm{E}\left[\min\left(T_L \mid T_S\right)\right]$ where T_L reflects the time alive (life expectancy) and T_S reflects the time in the susceptible state before becoming infected, which can be simplified (Anderson et al. (1992)) to:

$$\mathrm{E}\left[\min\left(T_L \mid T_S\right)\right] = \int_0^\infty e^{-(\mu+\delta)x} dx = \frac{1}{\mu+\delta}$$

such that the number of susceptible persons is the number entering the susceptible compartment μN times the duration of susceptibility:

$$S = \frac{\mu N}{\mu+\lambda}$$

Analogously, the steady-state number of infected persons is the number entering the infected state from the susceptible state (number susceptible, times rate of infection $\lambda = \frac{\beta I}{N}$, times the duration of infectiousness $\frac{1}{\mu+v}$

$$I = \frac{\mu N}{\mu+\lambda}\lambda\frac{1}{\mu+v}$$

2.1.3 | SEIR Model

Assuming that the incubation period is a random variable with exponential distribution with parameter a (i.e. the average incubation period is a^{-1}), and also assuming the presence of vital dynamics with birth rate Λ equal to death rate μ, we have the model: $\frac{dS}{dt} = \Lambda N - \mu S - \frac{\beta IS}{N}$ $\frac{dE}{dt} = \frac{\beta IS}{N} - (\mu+a)E$ $\frac{dI}{dt} = aE - (\gamma+\mu)I$ $\frac{dR}{dt} = \gamma I - \mu R$ We have $S+E+I+R = N$, but this

is only constant because of the simplifying assumption that birth and death rates are equal; in general N is a variable. For this model, the basic reproduction number is:

$$R_0 = \frac{a}{\mu + a} \frac{\beta}{\mu + \gamma}$$

Similarly to the SIR model, also, in this case, we have a Disease-Free-Equilibrium ($(N, 0, 0, 0)$ and an Endemic Equilibrium EE, and one can show that, independently from biologically meaningful initial conditions:
$(S(0), E(0), I(0), R(0)) \left\{(S, E, I, R) \in [0, N]^4 : S \geq 0, E \geq 0, I \geq 0, R \geq 0, S + E + I + R = N\right\}$

it holds that: $R_0 \leq 1 \Rightarrow \lim_{t \to +\infty}(S(t), E(t), I(t), R(t)) = DFE = (N, 0, 0, 0)$ $R_0 > 1, I(0) > 0 \Rightarrow \lim_{t \to +\infty}(S(t), E(t), I(t), R(t)) = EE$ In case of periodically varying contact rate $\beta(t)$ the condition for the global attractiveness of DFE is that the following linear system with periodic coefficients: $\frac{dE_1}{dt} = \beta(t) I_1 - (\gamma + a) E_1$

$$\frac{dI_1}{dt} = aE_1 - (\gamma + \mu) I_1$$

is stable.

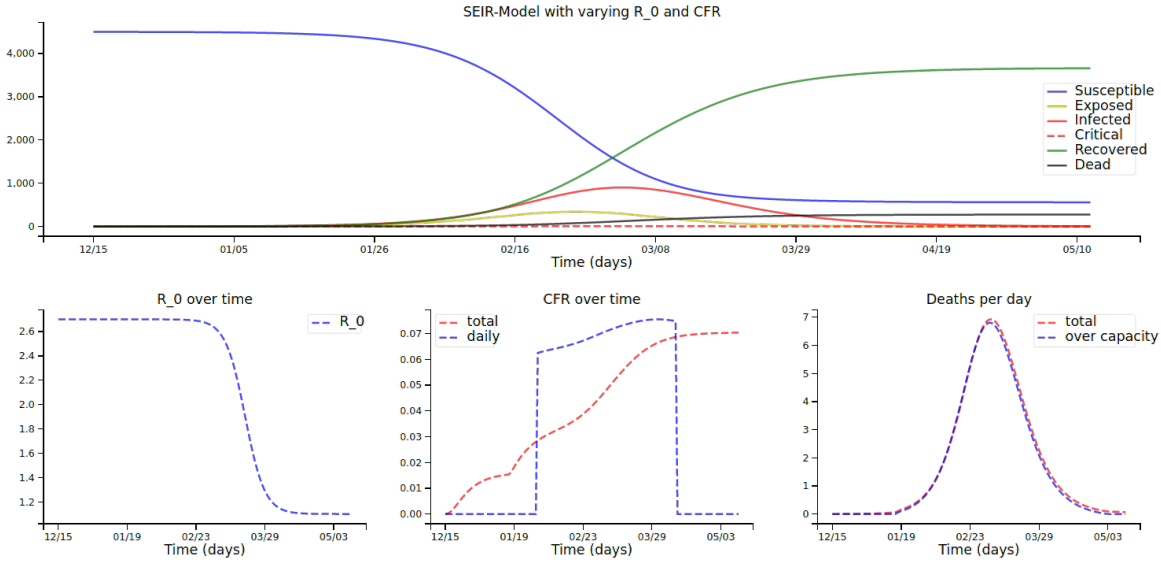

Figure 2.5: SEIR Graph

2.2 | MATLAB Overview

MATLAB is a proprietary multi-paradigm programming language and numerical computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of

functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

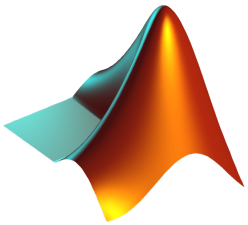

Figure 2.6: MATLAB Logo

2.2.1 | Functions

Both scripts and functions allow you to reuse sequences of commands by storing them in program files. Scripts are the simplest type of program, since they store commands exactly as you would type them at the command line. Functions provide more flexibility, primarily because you can pass input values and return output values. For example, this function named fact computes the factorial of a number (n) and returns the result (f).

```
function f = fact(n)
    f = prod(1:n);
end
```

This type of function must be defined within a file, not at the command line. Often, you store a function in its own file. In that case, the best practice is to use the same name for the function and the file (in this example, fact.m), since MATLAB® associates the program with the file name. Save the file either in the current folder or in a folder on the MATLAB search path.

You can call the function from the command line, using the same syntax rules that apply to functions installed with MATLAB. For instances, calculate the factorial of 5.

```
x = 5;
y = fact(5)
```

```
y =
    120
```

Starting in R2016b, another option for storing functions is to include them at the end of a script file. For instance, create a file named mystats.m with a few commands and two functions, fact

and perm. The script calculates the permutation of (3,2).

```
x = 3;
y = 2;
z = perm(x,y)

function p = perm(n,r)
    p = fact(n)*fact(n-r);
end

function f = fact(n)
    f = prod(1:n);
end
```

Call the script from the command line.

```
mystats
```

```
z =
   6
```

2.2.1.1 | Function Definition Syntax

The first line of every function is the definition statement, which includes the following elements.

- **function keyword** (required)
  Use lowercase characters for the keyword.
- **Function name** (required)
  Valid function names follow the same rules as variable names. They must start with a letter, and can contain letters, digits, or underscores.
- **Output arguments** (optional)
  If your function returns one output, you can specify the output name after the function keyword.

```
function myOutput = myFunction(x)
```

  If function returns more than one output, enclose the output names in square brackets.

```
function [one,two,three] = myFunction(x)
```

If there is no output, you can omit it.

```
function myFunction(x)
```

Or you can use empty square brackets.

```
function [] = myFunction(x)
```

- **Input arguments** (optional)
  If your function accepts any inputs, enclose their names in parentheses after the function name. Separate inputs with commas.

```
function y = myFunction(one,two,three)
```

  If there are no inputs, you can omit the parentheses.

2.2.2 | Plotting

There are many types of plots in MATLAB, as shown in figure 2.7

2.2.2.1 | 2D Plotting

plot(X,Y) creates a 2-D line plot of the data in Y versus the corresponding values in X.

- If X and Y are both vectors, then they must have equal length. The plot function plots Y versus X.
- If X and Y are both matrices, then they must have equal size. The plot function plots columns of Y versus columns of X.
- If one of X or Y is a vector and the other is a matrix, then the matrix must have dimensions such that one of its dimensions equals the vector length. If the number of matrix rows equals the vector length, then the plot function plots each matrix column versus the vector. If the number of matrix columns equals the vector length, then the function plots each matrix row versus the vector. If the matrix is square, then the function plots each column versus the vector.
- If one of X or Y is a scalar and the other is either a scalar or a vector, then the plot function plots discrete points. However, to see the points you must specify a marker symbol, for example, plot(X,Y,'o').

```
% EXAMPLE 1: BASIC 2D PLOTTING

x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = sin(x-0.25);
y3 = sin(x-0.5);

figure
plot(x,y1,x,y2,'--',x,y3,':')
```

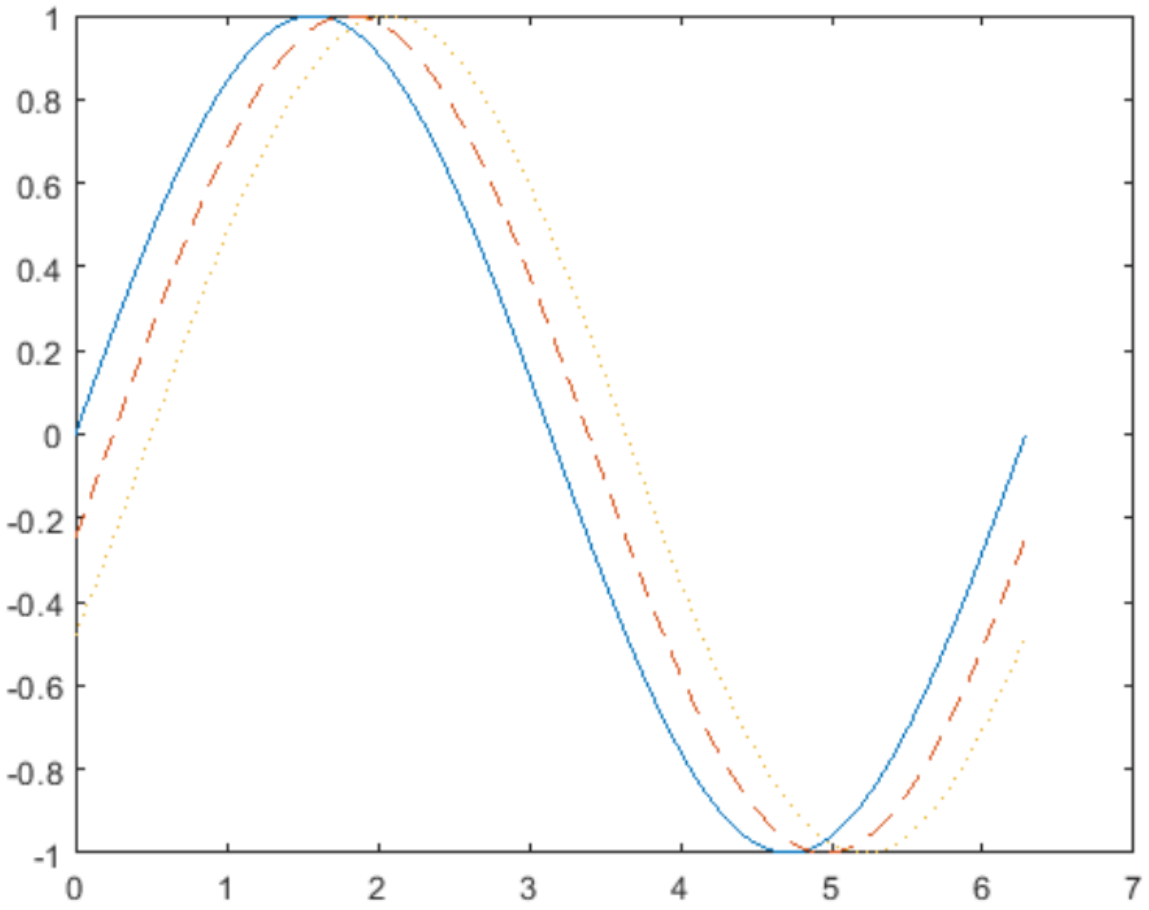

Figure 2.8: Example 1: Basic 2D Plotting

```
% EXAMPLE 2: PLOT FITTING
n=1:8;
x1(n)=[1 2 3 4 5 6 7 8];
```

```
y1(n)=[1 2 3.5 4.5 5 5.7 6.8 8];
plot(n,x1,'o',n,y1,'o','linewidth',2)
title('data points')
xlabel('n')
ylabel('x1 and y1')
grid on

A1=polyfit(x1,y1,1)
xi=linspace(1,8,50);
yi=polyval(A1,xi);
plot(x1,y1,'o',xi,yi)
title('polyfit')
xlabel('x1 and xi')
ylabel('y1 and yi')
grid on
A7=polyfit(x1,y1,7)
xi=linspace(1,8,50);
yi=polyval(A7,xi);
plot(x1,y1,'o',xi,yi)
title('polyfit')
xlabel('x1 and xi')
ylabel('y1 and yi')
grid on
```

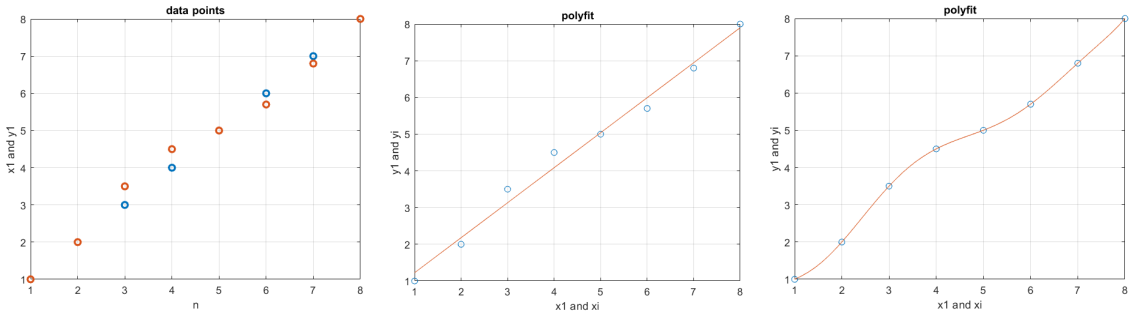

Figure 2.9: Example 2: 2D Plot Fitting

2.2.2.2 | 3D Plotting

- Mesh Plot

  The *mesh* function creates a wireframe mesh. By default, the color of the mesh is propor-

tional to the surface height.

- Surface Plot
  The *surf* function is used to create a 3-D surface plot.

Listing 2.1: Example: 3D Gaussian Plot

```
% EXAMPLE: 3D GAUSSIAN FUNCTION FLOT

a=1; % height of peak
b=1; % position of peak center
c=1; % standard deviation
x=-10:0.1:10;
y=x;

[X,Y]=meshgrid(x,y);
g3d=exp(-(X.^2+Y.^2));
surf(x,y,g3d)
xlim([-5 5])
ylim([-5 5])
zlim([0 1])
```

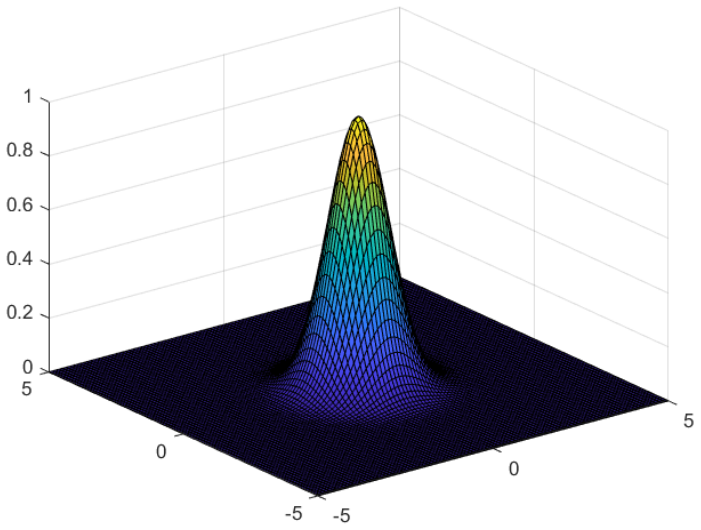

Figure 2.10: 3D Gaussian Function

2.2.3 | App Designer

App Designer lets you create professional apps without having to be a professional software developer. Drag and drop visual components to lay out the design of your graphical user interface (GUI) and use the integrated editor to quickly program its behavior. The corresponding files are saved in the *.mlapp* format.

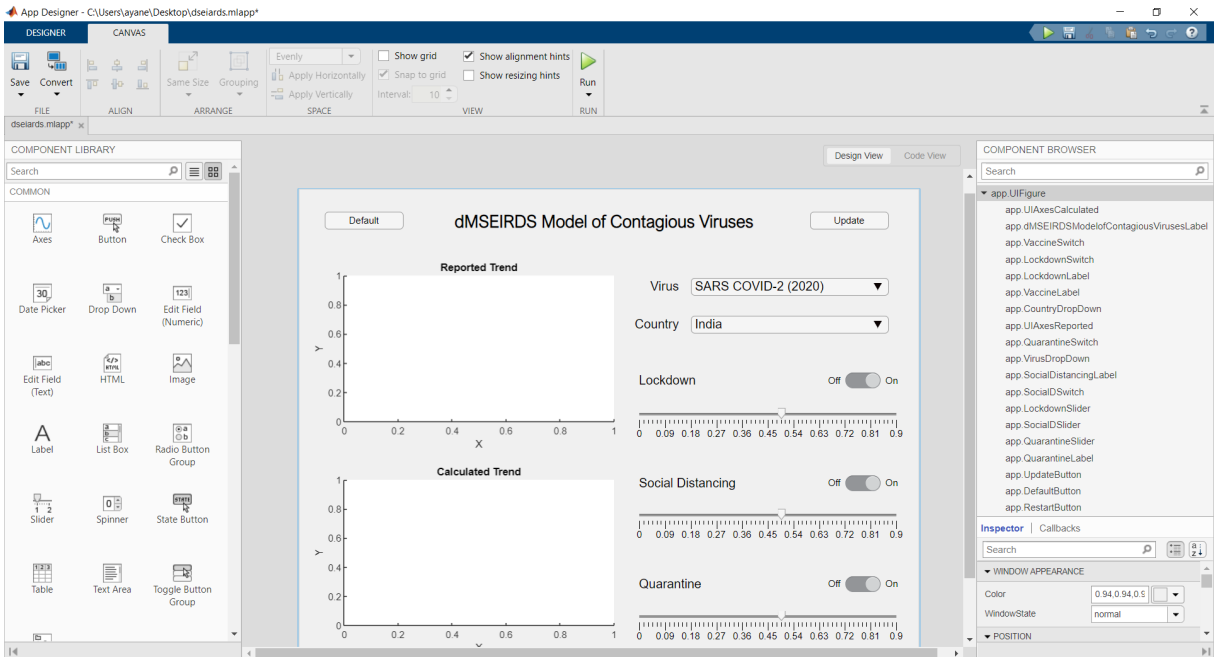

Figure 2.11: Screenshot of our first model plan, changed later

2.2.3.1 | Design a User Interface

Drag and drop visual components to the design canvas and use alignment hints to get a precise layout. App Designer automatically generates the object-oriented code that specifies the app's layout and design.

2.2.3.2 | Define App Behavior

Use the integrated version of the MATLAB Editor to define your app's behavior. App Designer can automatically check for coding problems using the Code Analyzer. You can view warning and error messages about your code as you're writing it, and modify your app based on the messages. You can also model the app behavior using a Stateflow® chart.

Figure 2.7: Types of Plots in MATLAB

3

Methodology

3.1 | Compartment Modification

The model we use as the base line for our model is the $SEIRD$ model, a step ahead from $SEIR$ which describes the epidemic trend fairly accurately. We first came across this model in Fonseca i Casas et al. (2020) which describes the $SEIRD$ using multiple diagrams but the main jist of the model is given by 3.1.

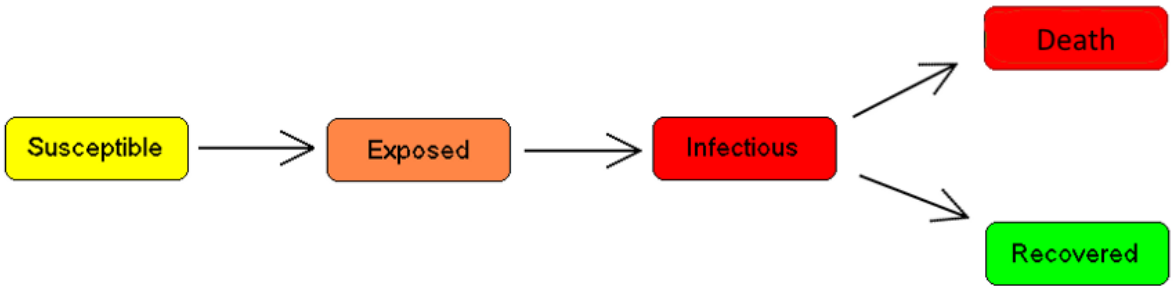

Figure 3.1: Diagrammatic view of the SEIRD model - from Fonseca i Casas et al. (2020)

Figure 3.2 is the diagrammatic view of the compartmental view of our model, which includes an additional L and Q compartment which stand for

- L $\rightarrow$ Lockdown
- Q $\rightarrow$ Quarantine

as discussed in earlier sections. As visible in the figure, there are three sinks (L, R and D) in our $SEIRD - LQ$ model compared to the two sinks in the $SEIRD$ model (R and D). Sinks here mean that the population can be contained in those compartments without the requirement of them moving onto another compartment. One way to identify sinks is the one-way flow of population - they can come into the category/compartment, but not go out of it. The *death* compartment is a great example as it would be not just mathematically, but terrifyingly wrong

if someone were able to come back out from the *death* compartment. Hence D is a sink, in case of both $SEIRD$ and $SEIRD - LQ$.

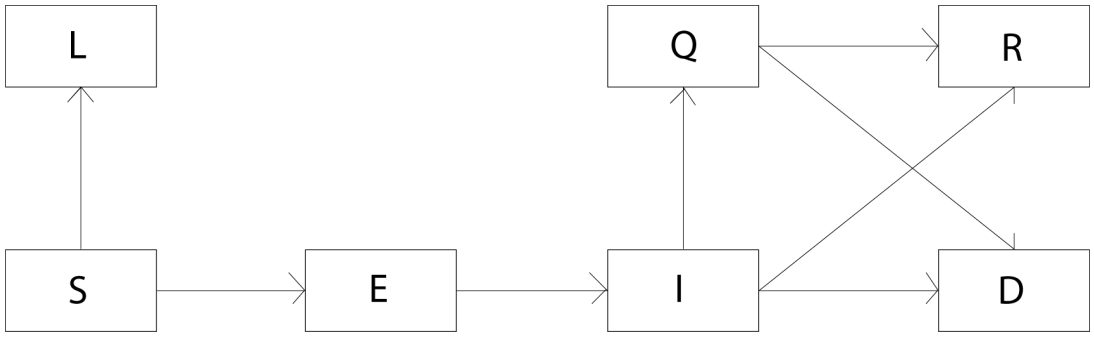

Figure 3.2: Diagrammatic view of our $SEIRD - LQ$ model - made in *Adobe Illustrator*

3.2 | Mathematical Tweaking

3.2.1 | SEIRD Model

Let us take over each compartment of the SEIRD model one by one and represent its change mathematically.

$$S'(t) = -\frac{\beta S(t) I(t)}{N} \tag{3.1}$$

$$E'(t) = \frac{\beta S(t) I(t)}{N} - \alpha E \tag{3.2}$$

$$I'(t) = \alpha E(t) - \gamma I(t) \tag{3.3}$$

$$R'(t) = \gamma(1-\mu) I(t) \tag{3.4}$$

$$D'(t) = \gamma \mu I(t) \tag{3.5}$$

$$N = S + E + I + R + D \tag{3.6}$$

3.2.2 | SEIRD-LQ Model

Let us take over each compartment of the SEIRD-LQ model one by one and represent its change mathematically.

$$\frac{dS}{dt} = -\beta \left(1 - C_Q\right) \left(1 - C_{SD}\right) IS - C_L S \tag{3.7}$$

$$\frac{dE}{dt} = \beta\,(1 - C_Q)\,(1 - C_{SD})\,IS - \sigma E \tag{3.8}$$

$$\frac{dI}{dt} = \sigma E - (1 - C_Q)\,\gamma I - C_Q \gamma I \tag{3.9}$$

$$\frac{dR}{dt} = (1 - \Delta)\gamma I \tag{3.10}$$

$$\frac{dD}{dt} = \Delta \gamma I \tag{3.11}$$

$$S + E + I + R + D + L = N \tag{3.12}$$

3.3 | Data Collection

For this section, we looked not just into research papers but also the WHO website and other trusted health organization sites.

3.3.1 | Countries (C)

In our model, we are currently taking into account, four countries

Countries	Population (N)	Hospital Capacity (Hosp)
India	1.353e9	0.53
Russia	144.5e6	8.05
UAE	9.631e6	1.4
USA	328.2e6	2.77

3.3.2 | Viruses (V)

In our model, we are currently taking into account, two viruses

- SARS-COVID 2 (2020)
- Ebola (2014)

In case of Ebola, as the outbreak was mainly only bounded by Africa. We take global averages for datasets that do not have enough backup and/or data points.

Virus	Reproduction Rate	Death Rate	Incubation Period	Infectious Period
SARS COVID-2	2.2-2.9	1.5-2.5%	4.2	14
Ebola	1.8-2.1	30-90%	9	20

3.3.3 | Cross Product of C and V

India	N	Hosp	R_0	Death	T_Inc	T_Inf
SARS COVID-2	1.353e9	0.53	2.9	1.49%	4.2	14
Ebola	1.353e9	0.53	1.8	50%	9	20

Russia	N	Hosp	R_0	Death	T_Inc	T_Inf
SARS COVID-2	144.5e6	8.05	2.2	1.72%	4.2	14
Ebola	144.5e6	8.05	1.8	50%	9	20

UAE	N	Hosp	R_0	Death	T_Inc	T_Inf
SARS COVID-2	9.631e6	1.4	2.8	0.37%	4.2	14
Ebola	9.631e6	1.4	1.8	50%	9	20

USA	N	Hosp	R_0	Death	T_Inc	T_Inf
SARS COVID-2	328.2e6	2.7	2.2	4%	4.2	14
Ebola	328.2e6	2.7	1.8	50%	9	20

3.4 | Simulating the Model

The code written by us for our $SEIRD - LQ$ model has been shown in Appendix B and is the second main work of our project. Please click on the hyperlink in previous sentence in blue, in order to view the code. Below is a screenshot 3.3 showing the outline of our application. There are more screenshots of our application at work, discussing the visual features and results derived from the application.

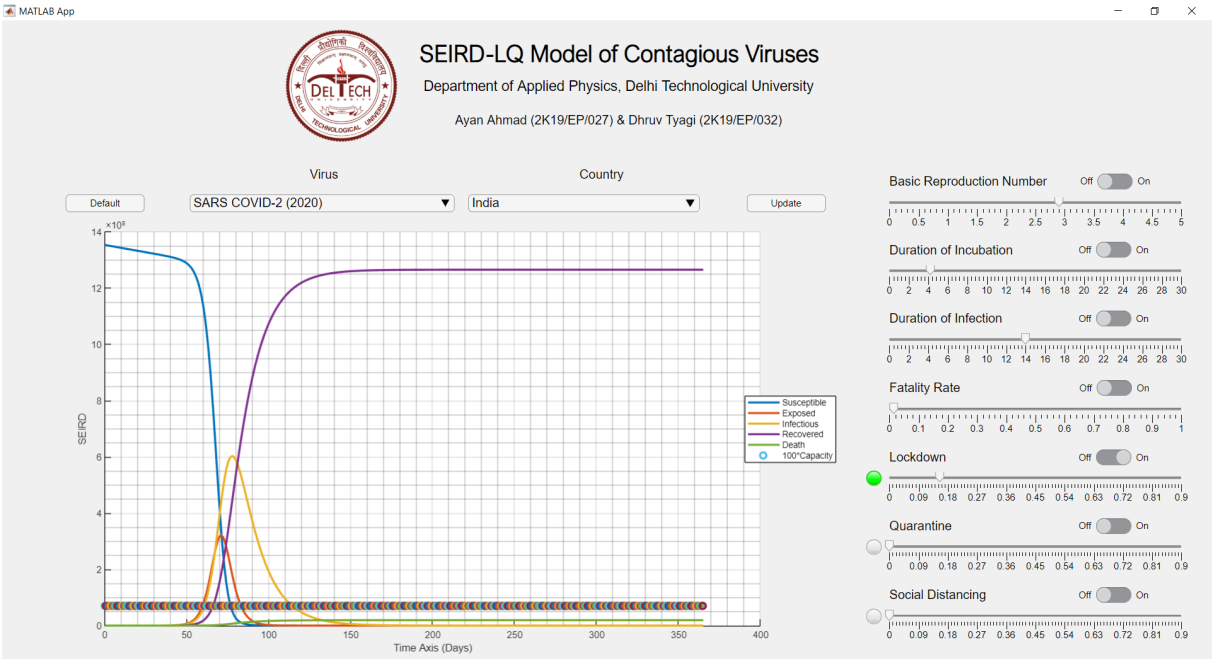

Figure 3.3: A look at the our SEIRD-LQ Application

4

Results

In this section we will be discussing the functionality of each component

- Buttons
- Switches
- Sliders
- Dropdown Menus

via the captions of the screenshots. 4.1 is the basic outline of our application. The screen the user will see on its launch.

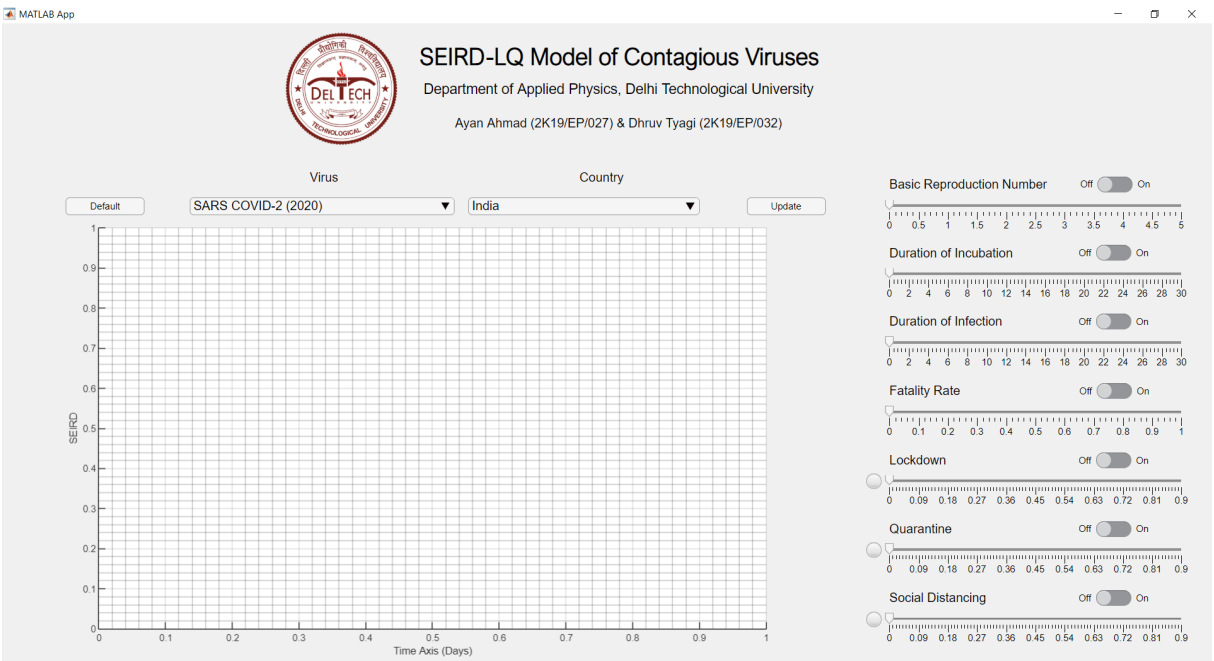

Figure 4.1: Basic outline of the application, startup screen

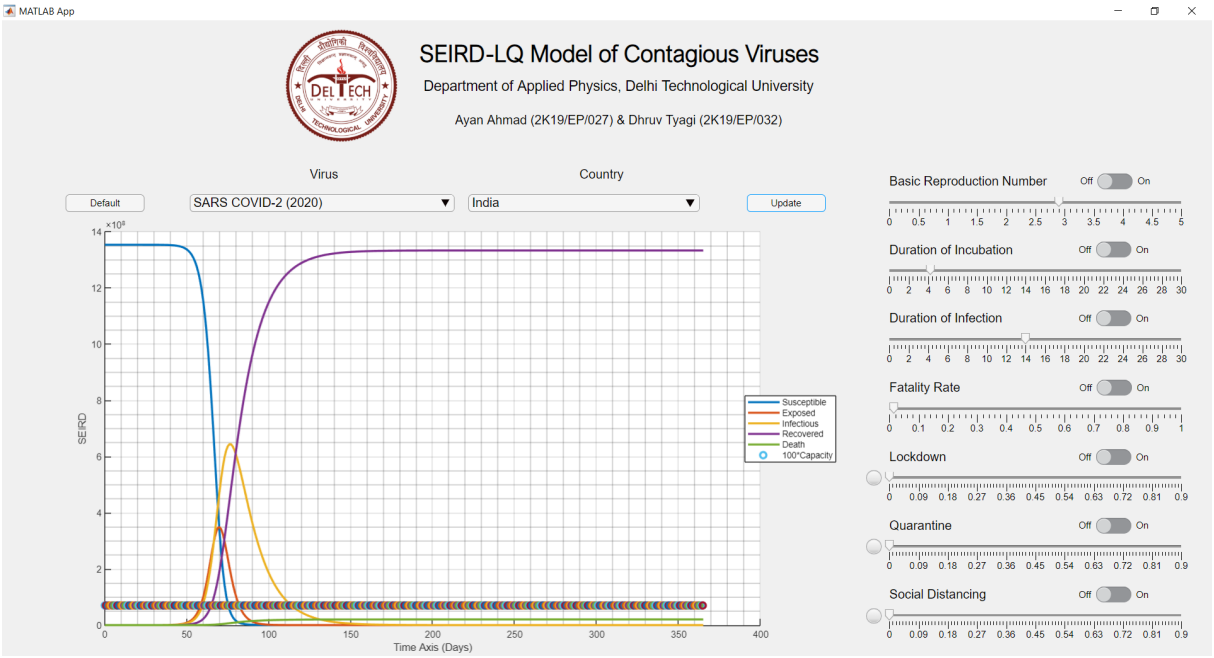

Figure 4.2: The first run requires the user to press the update button, the default values are assigned to each slider and the epidemic trend is plotted.

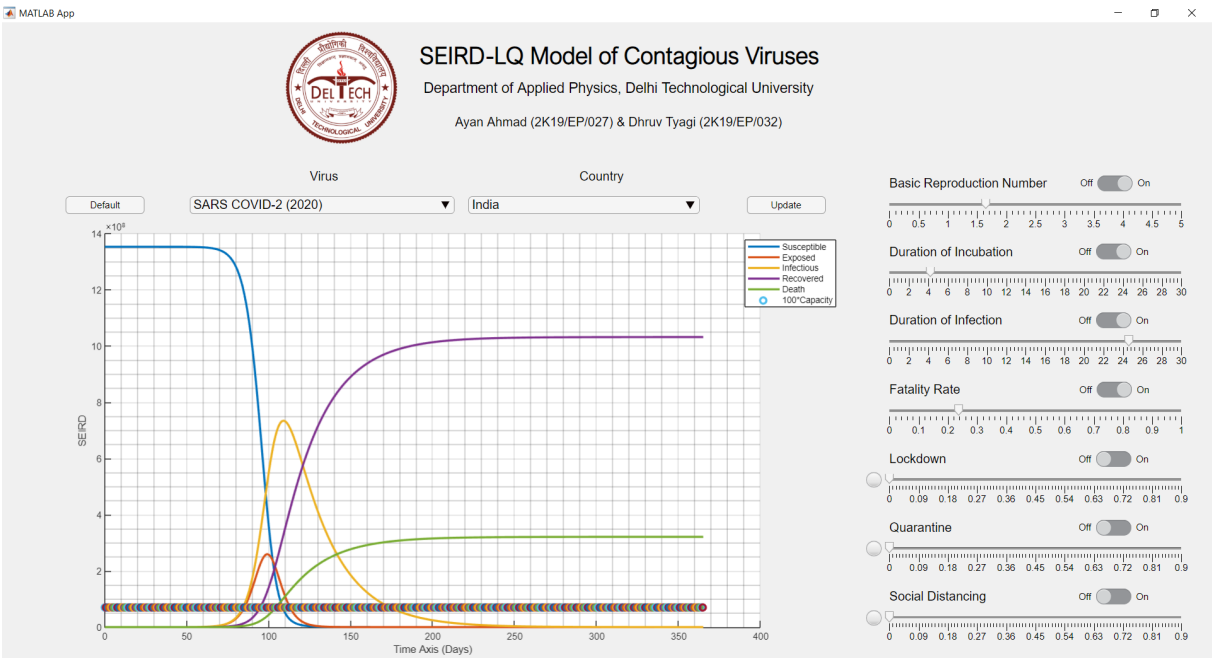

Figure 4.3: The user can switch on the parameter switches in order to be able to change them to their will and see how the curves depend on each parameter in our model.

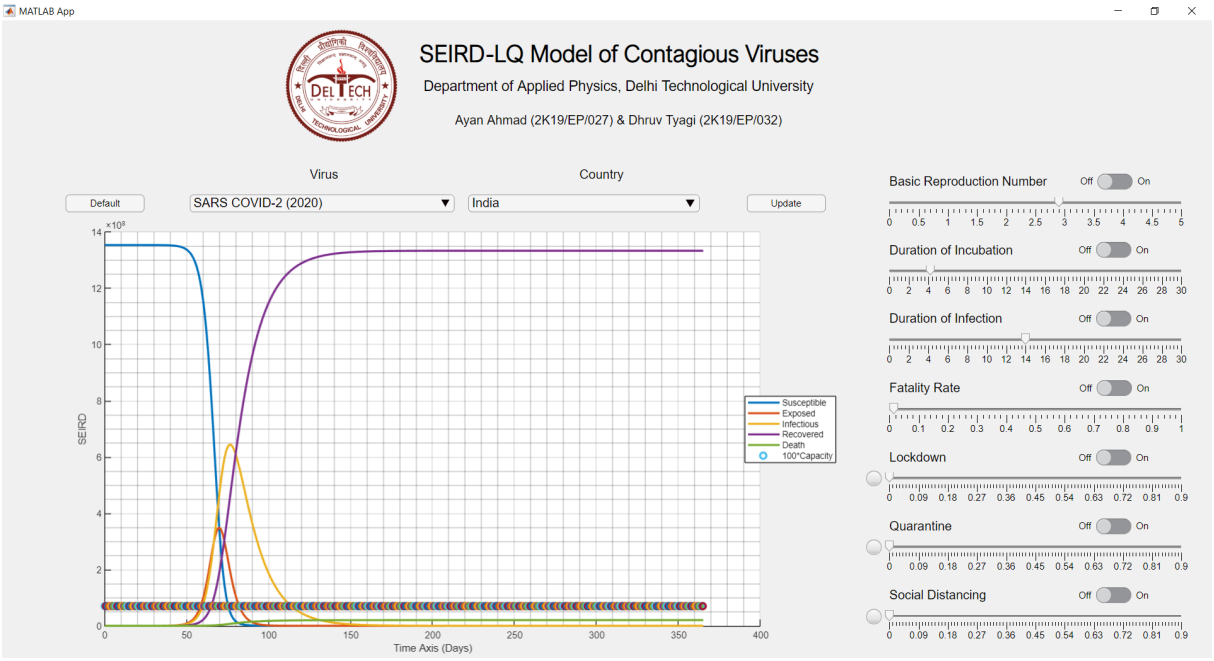

Figure 4.4: On pressing the default, and the update button, all values (sliders and switches) will go back to their original configuration for the particular dropdown combination.

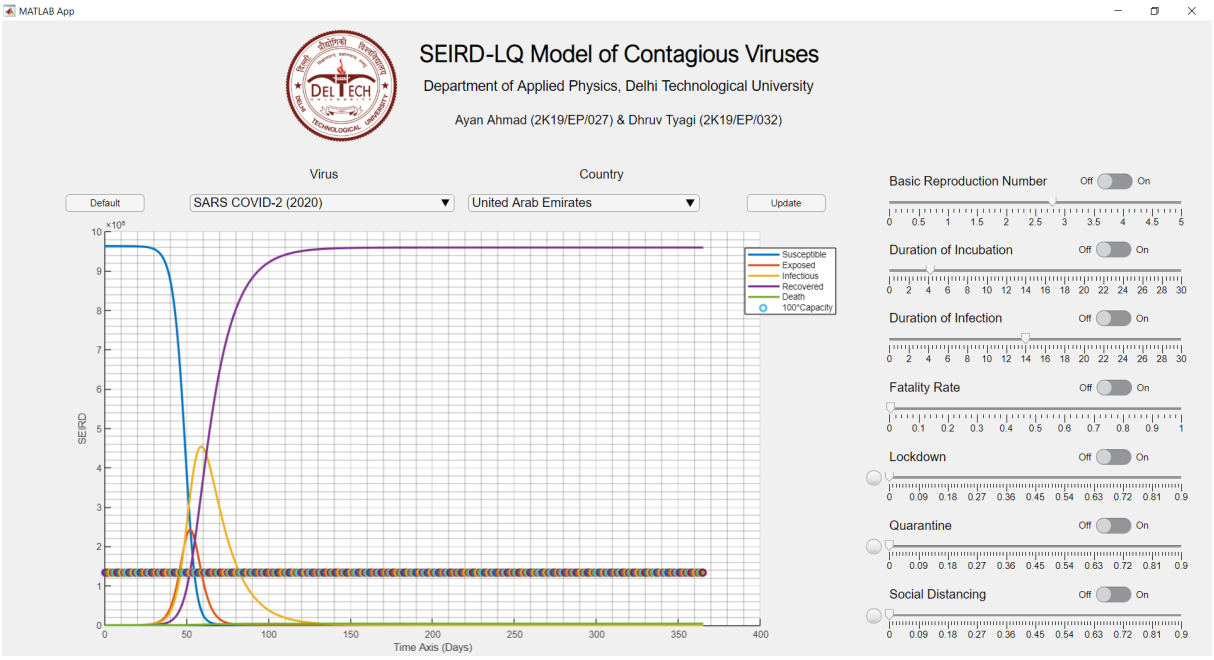

Figure 4.5: The user can select a different country from the country dropdown list. The values will be reset to default for the new combination (stored in code).

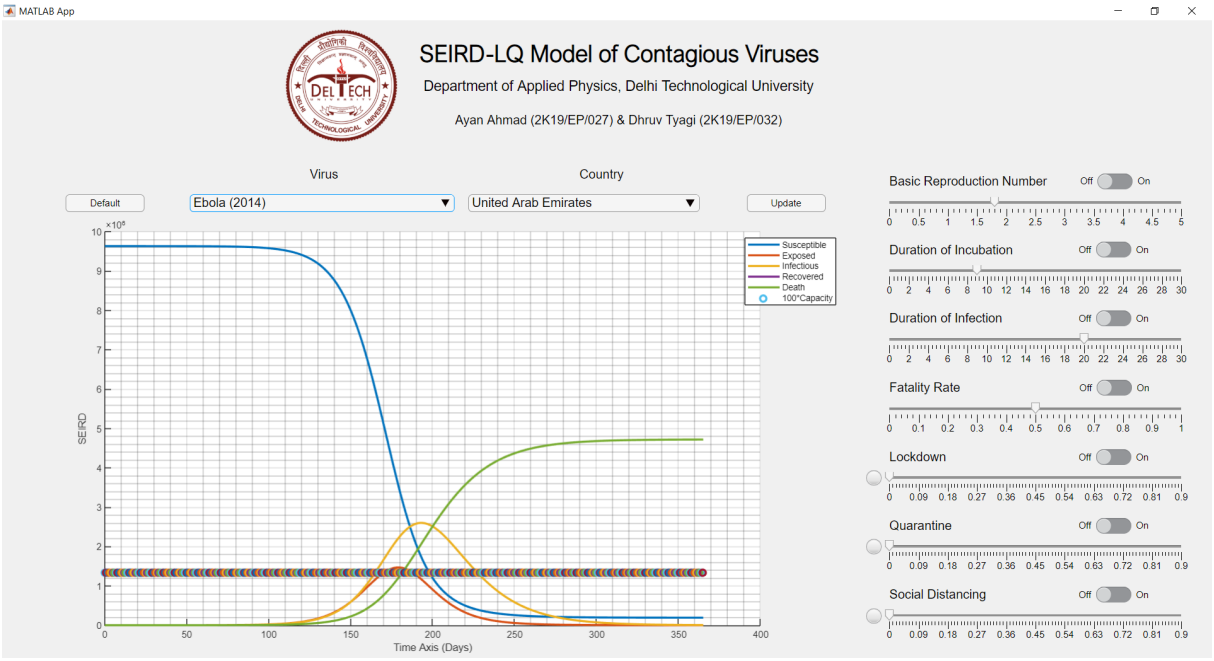

Figure 4.6: The user can select a different virus from the virus dropdown list. The values will be reset to default for the new combination (stored in code).

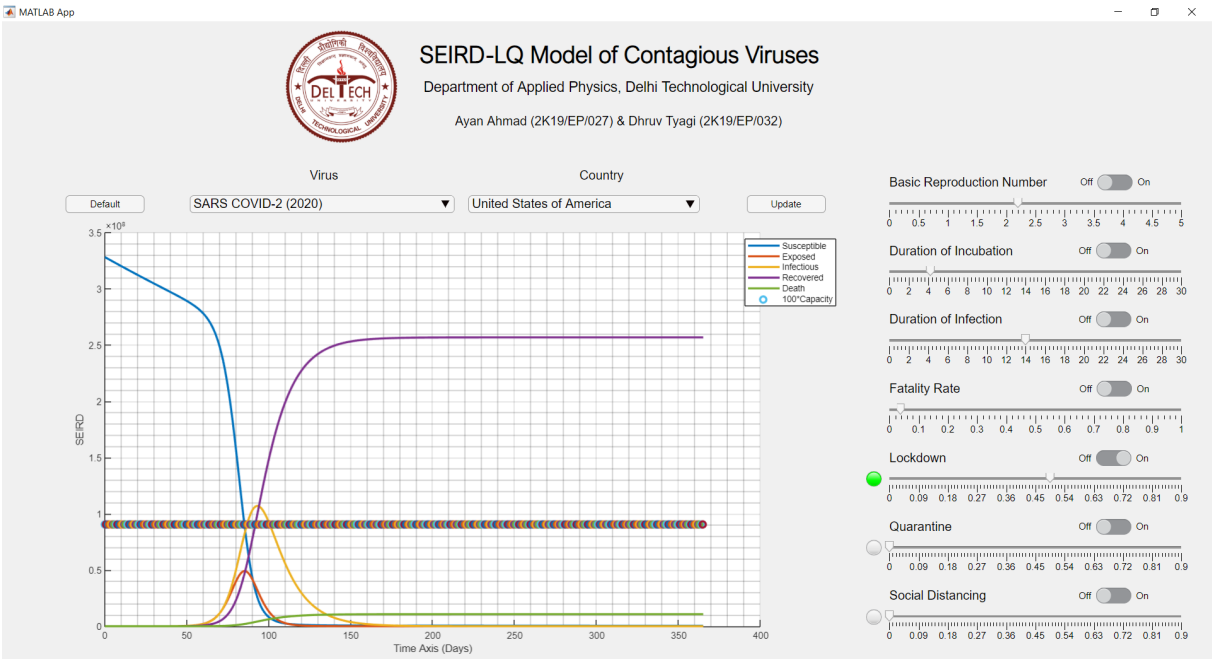

Figure 4.7: This is the change is curves when we set the lockdown quality to roughly 0.5 while keeping other values to their default. The lamp beside lockdown slider goes green.

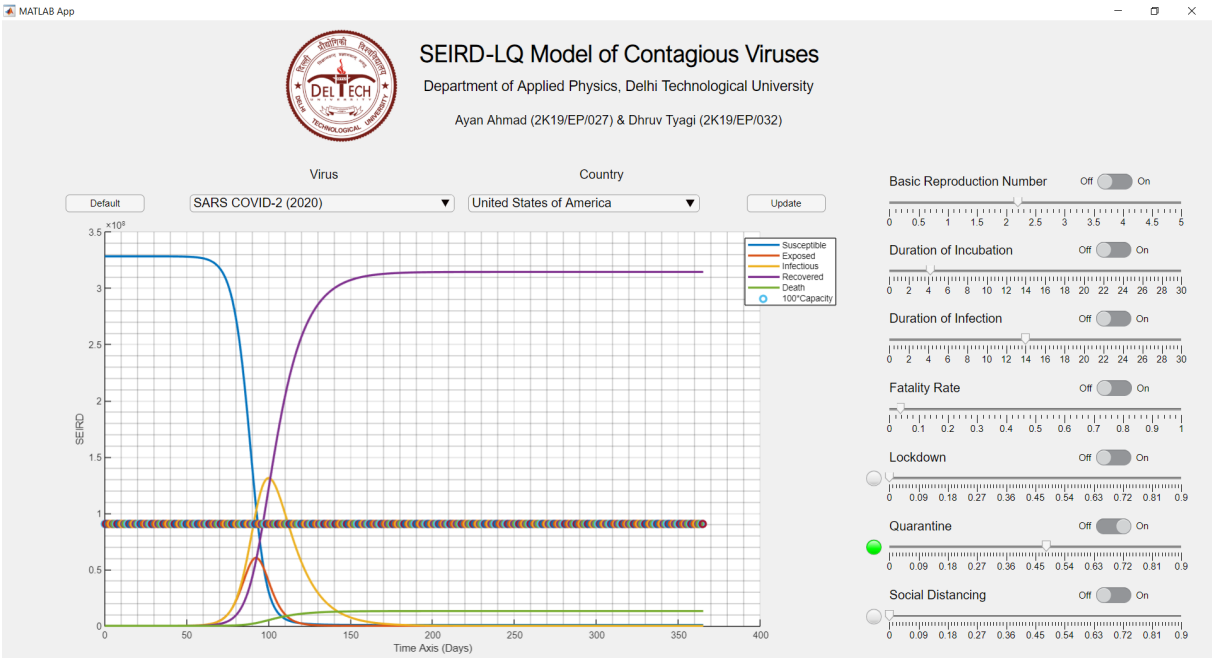

Figure 4.8: This is the change is curves when we set the quarantine quality to roughly 0.5 while keeping other values to their default. The lamp beside quarantine slider goes green.

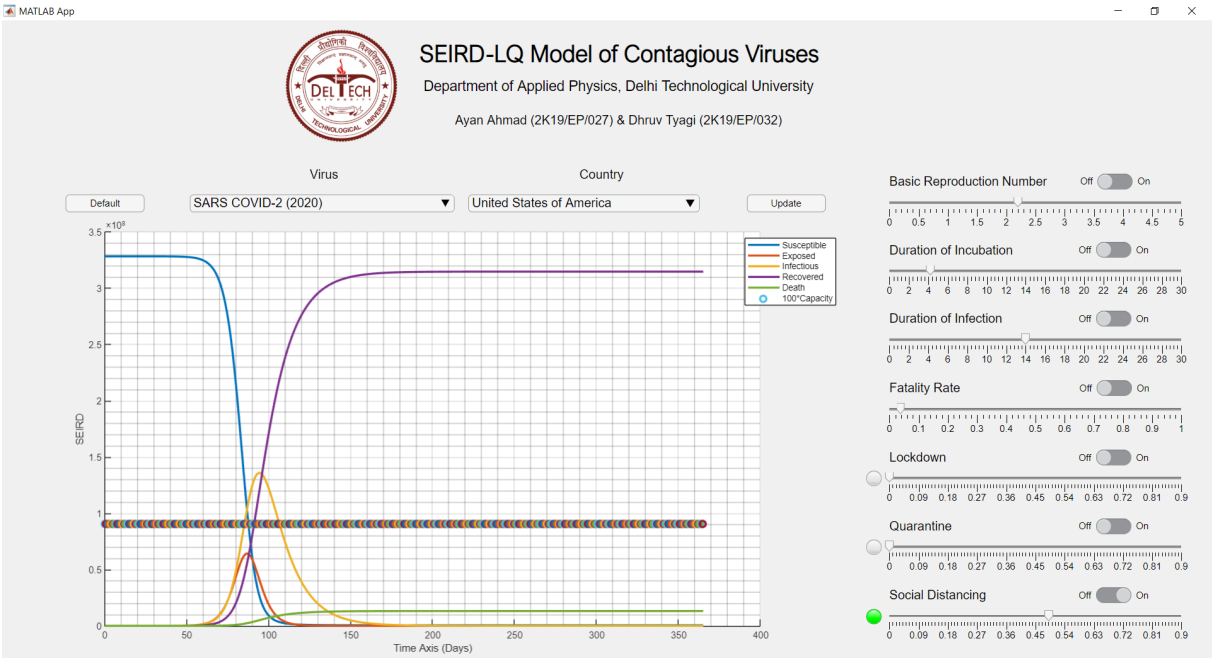

Figure 4.9: This is the change is curves when we set the social distancing quality to roughly 0.5 while keeping other values to their default. The lamp beside distancing slider goes green.

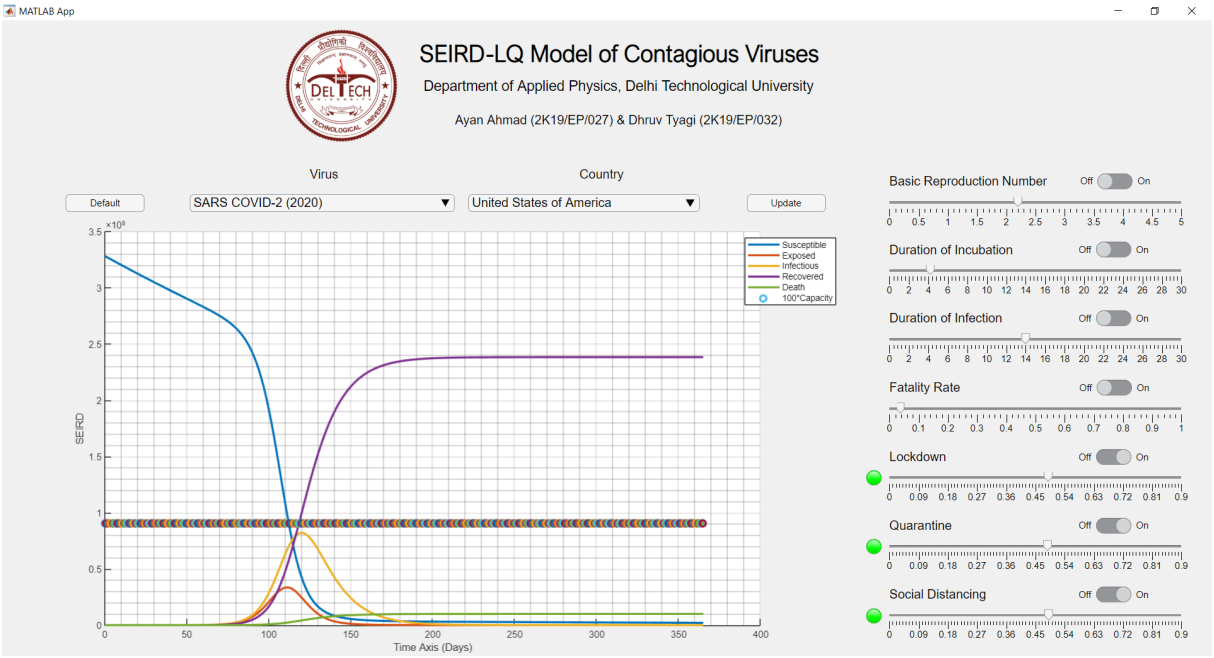

Figure 4.10: This is the change is curves when we set the all prevention quality to roughly 0.5 while keeping other values to their default. The three lamps all go green.

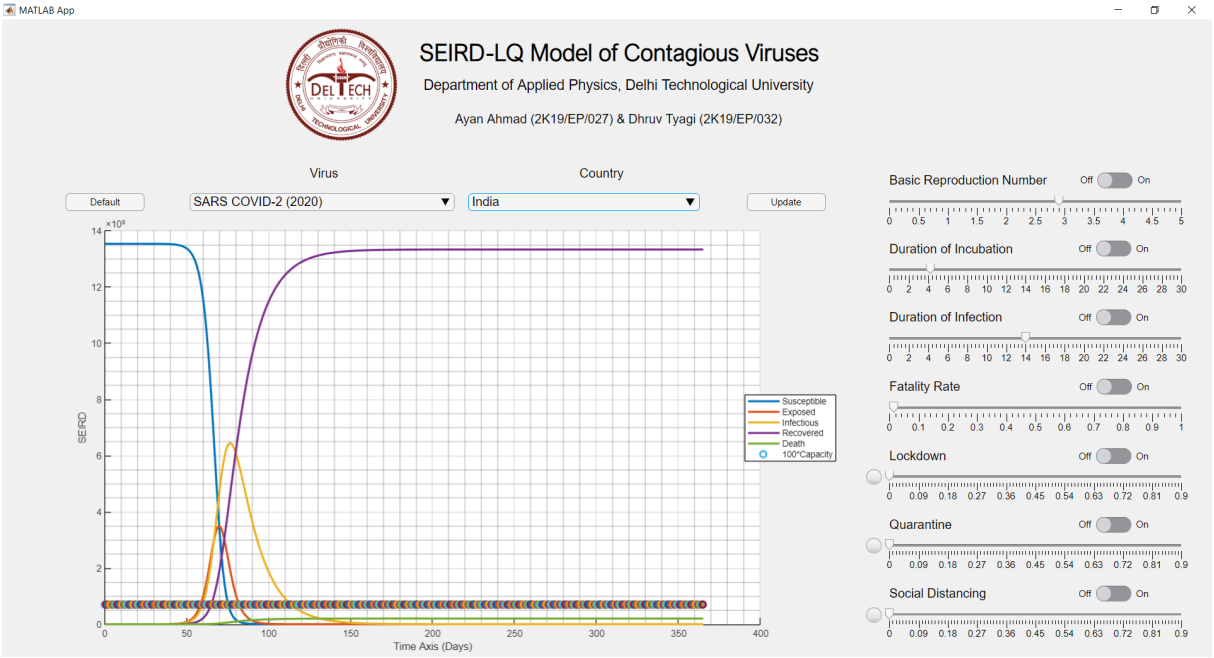

Figure 4.11: Choosing another combination form the dropdown menus automatically sets the values to new default rather than keeping it how it was.

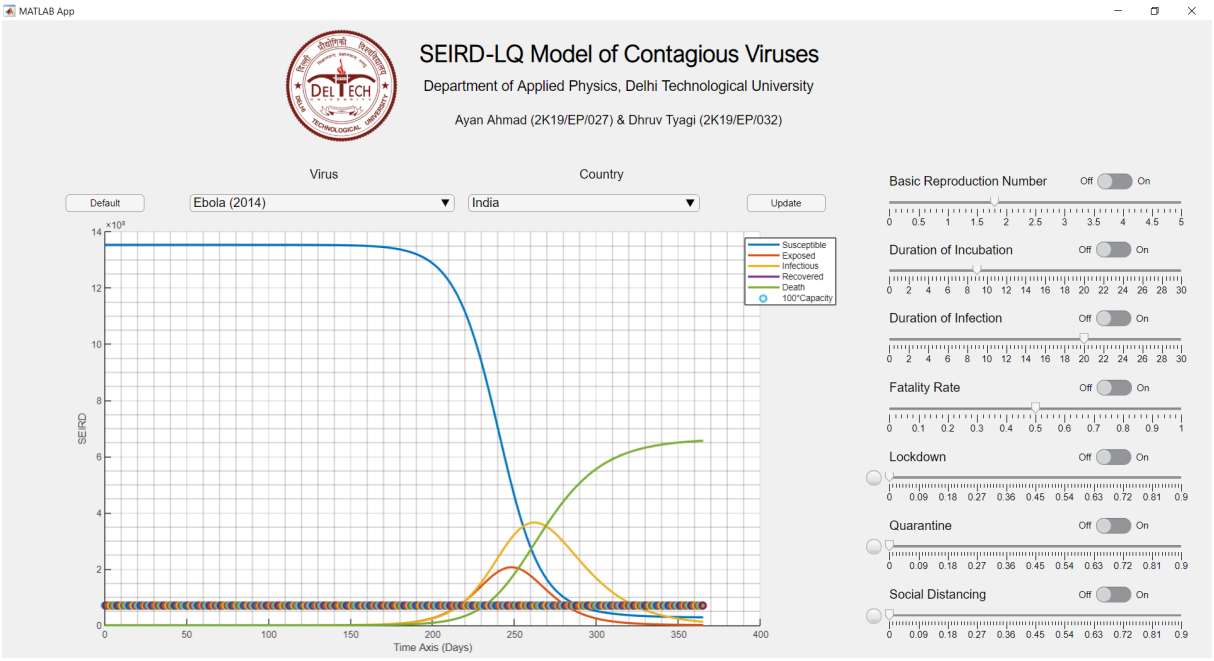

Figure 4.12: For the same combinations and prevention, Ebola causes less infections than Covid 2020.

5

Discussion & Conclusions

Besides the basic observations of other models, following are the conclusions of our application and modified model

5.0.1 | Effective methods for flattening the curve

Lockdown is the most effective method to flatten the curve. The order of effectiveness in terms of flattening the curve in order from greatest to lowest is as follows (for CoVid 2020)

- Lockdown
- Quarantine
- Social Distancing

5.0.2 | Position and arrival of peak

Smaller populations can result in an early peak as the graphs will have different proportion. The proportion can be described as the length of y axis in the plot and the peak value. For a country with lower population, the peak is expected to arise early assuming same contact density.

5.0.3 | Incubation and Infectious Period

For constant counterpart, T_{inc} is inversely and T_{inf} is directly proportional to the infectious peak. This makes sense as longer the incubation time per infection time, lesser the number of exposed people at every point in time. And longer the infectious duration, higher the chance of exposure and transmission as the infection lasts longer for every person contracting it.

5.0.4 | Fatality rate and death count

Higher fatality rate does not necessarily mean higher death count. Although the fatality rate for Ebola was high and it did cause higher deaths in our model. It is not impossible in our model to image a virus that can have that perfect match of contagion and fatality rate so as to decrease death toll by

- Being deadly but not very contagious
- Being highly deadly and medium-level contagious
- Not being deadly but being contagious of any level

6

Future Work

During our time working on the project, we have realised that we've just seen the tip of the iceberg and there's so much more to epidemics and MATLAB than our current knowledge holds. So as a part of our future work, we have two proposals to suggest and work on beyond the span of the project deadline.

6.1 | Library of Contagious Viruses

We have purposefully designed our code and interface with dropdown lists for country and virus so that we can collection more data as it is published and effortlessly be able to add the constants of model to the list of countries (which determine R_0, N, $Death_{Rate}$) and viruses (which determine R_0, $Death_{Rate}$, T_{inc}, T_{inf}) in order to output specific trends in the epidemic.

6.2 | Addition of Parameters and Compartments

It is no surprise that any model, be it $SEIRD$ or our $SEIRD - LQ$, do not accurately describe the reported trend for a long enough time interval. This is because on a microscopic scale, there are many other factors that come into play such as asymptotic cases where the person is less likely to be quarantined and so on. If we were to take into account every small detail, we would have too many parameter constants in our equations which would need a lot of data, that too on a microscopic level, which can be difficult considering we must distance ourselves as well. So whenever there is enough data available on the virus in a particular microregion, a parameter can be added to the equation in order to achieve more accurate, non-ideal results.

6.3 | Mathematical Analysis

A more rigorous mathematical analysis and formulation of models for epidemiology.

6.4 | Localization

To dissect the countries further into smaller regions for more accurate data on microscopic changes and more accurate model.

A

MATLAB SIR Code with Gillespieal Algorithm (Literature Review)

Listing A.1: Gillespieal Algorithm for SIR Model - MATLAB Code

```
% SIRS EPIDEMIC MODEL - LITERATURE REVIEW

% This is not an original code, it had been analyzed by us to better
   understand epidemic simulating techniques in MATLAB. Our original
   (SEIRD-LQ) code is in the next appendix. The credit for this
   particular code goes to Stochastic simulation of epidemics.

clear all
for kk=1:2:3
    gam=0.1;alpha=.05;delta=.01; beta=0.3; i0=2; % Parameters
    if kk==1
        N=100
        time=120;
    else
        N=500
        time=300;
    end

%Eulers method to solve ODE

% Initial values
dt=.05; tim=time/dt;
i(1)=i0; s(1)=N-i0;r(1)=0;
ssum=0;
for tt=1:tim
    nn=s(tt)+i(tt)+r(tt);
    i(tt+1)=i(tt)+dt*((beta/nn)*i(tt)*s(tt)-gam*i(tt)-alpha*i(tt));
    s(tt+1)=s(tt)+dt*(-(beta/nn)*i(tt)*s(tt)+delta*r(tt));
    r(tt+1)=r(tt)+dt*(gam*i(tt)-delta*r(tt));
```

```
    ssum=ssum+dt*(beta/nn)*i(tt)*s(tt);
end
TotalCases=round(ssum)+i0 % Cumm Cases up to time

subplot(2,2,kk)
plot([0:dt:time],i, k --  ,  linewidth  ,2);
xlabel(  Time  );
ylabel(  Infectives  );
axis([0,time,0,.3*N])
hold on

subplot(2,2,kk+1)
plot([0:dt:time],i, k --  ,  linewidth  ,2);
xlabel(  Time  );
ylabel(  Infectives  );
axis([0,20,0,20]);
hold on

for k=1:4 % Sample paths for MC, Gillespie algorithm
    clear t s i r
    t(1)=0; i(1)=i0; s(1)=N-i0;r(1)=0;
    j=1;
    while i(j)>0 & t(j)<time % Stop hits zero or at time
        nn=s(j)+i(j)+r(j);
        u1=rand;u2=rand;
        den=((beta/nn)*i(j)*s(j)+gam*i(j)+alpha*i(j)
        +delta*r(j));
        t(j+1)=-log(u1)/den+t(j) % Time to next event
        e1=(beta/nn)*s(j)*i(j)/den;
        e2=e1+gam*i(j)/den;
        e3=e2+alpha*i(j)/den;
        e4=e3+delta*r(j)/den;
        if (u2<=e1)
        s(j+1)=s(j)-1;
        i(j+1)=i(j)+1;
        r(j+1)=r(j);
        elseif (u2>e1 & u2<=e2)
        s(j+1)=s(j);
```

```
65          i(j+1)=i(j)-1;
66          r(j+1)=r(j)+1;
67          40 A MatLaB Programs
68          elseif (u2>e2 & u2<=e3)
69          s(j+1)=s(j);
70          i(j+1)=i(j)-1;
71          r(j+1)=r(j);
72          else
73          s(j+1)=s(j)+1;
74          i(j+1)=i(j);
75          r(j+1)=r(j)-1;
76      end
77      j=j+1;
78  end
79
80  if k==1
81      subplot(2,2,kk)
82      stairs(t,i, y -    ,    linewidth    ,2);
83      hold on
84      subplot(2,2,kk+1)
85      stairs(t,i, y -    ,    linewidth    ,2);
86      hold on
87  end
88  if k==2
89      subplot(2,2,kk)
90      stairs(t,i, b -    ,    linewidth    ,2);
91      hold on
92      subplot(2,2,kk+1)
93      stairs(t,i, b -    ,    linewidth    ,2);
94      hold on
95  end
96      if k==3
97      subplot(2,2,kk)
98      stairs(t,i, g -    ,    linewidth    ,2);
99      hold on
100     subplot(2,2,kk+1)
101     stairs(t,i, g -    ,    linewidth    ,2);
102     hold on
```

```
103 end
104 if k==4
105     subplot(2,2,kk)
106     stairs(t,i,  r  -    ,    linewidth    ,2);
107     hold off
108     subplot(2,2,kk+1)
109     stairs(t,i,  r  -    ,    linewidth    ,2);
110     hold off
111 end
112 end
113
114 % Estimate Probability of Epidemic Extinction
115 count=0;tots=10000;
116 for k=1:tots % Number of sample paths
117     clear t s i r
118     i=i0; s=N-i0; r=0;
119     j=1;
120     outb=min(25,.25*N);
121     sumi=i0;
122     while i>0 & sumi<outb %Stop hits zero or cumm cases=outb
123         u1=rand; u2=rand;
124         nn=s+i+r;
125         den=((beta/nn)*i*s+gam*i+alpha*i+delta*r);
126         e1=(beta/nn)*s*i/den;
127         e2=e1+gam*i/den;
128         e3=e2+alpha*i/den;
129         e4=e3+delta*r/den;
130         if (u2<=e1)
131         i=i+1;
132         s=s-1;
133         r=r;
134         sumi=sumi+1;
135         elseif (u2>e1 & u2<=e2)
136         s=s;
137         i=i-1;
138         r=r+1;
139         elseif (u2>e2 & u2<=e3)
140         s=s;
```

```
141         i=i-1;
142         r=r;
143         else
144         s=s+1;
145         i=i;
146         r=r-1;
147         end
148         j=j+1;
149         end
150         if i==0
151         count=count+1;
152     end
153 end
154 probextSP=count/tots % Ext Approx from Sample Paths
155 probextBP=((gam+alpha)/beta)^(i0) % Ext Est from BP
156 end
```

B

MATLAB Code for SEIRD-LQ App (Methodology)

```
classdef SEIRD_LQ < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        UIAxesCalculated                matlab.ui.control.UIAxes
        SEIRDLQModelofContagiousVirusesLabel  matlab.ui.control.Label
        CountryDropDown                 matlab.ui.control.DropDown
        VirusDropDown                   matlab.ui.control.DropDown
        UpdateButton                    matlab.ui.control.Button
        DefaultButton                   matlab.ui.control.Button
        AyanAhmad2K19EP027DhruvTyagi2K19EP032Label  matlab.ui.control
           .Label
        Image                           matlab.ui.control.Image
        DepartmentofAppliedPhysicsDelhiTechnologicalUniversityLabel
           matlab.ui.control.Label
        SocialDSwitch                   matlab.ui.control.Switch
        SocialDistancingLabel           matlab.ui.control.Label
        SocialDSlider                   matlab.ui.control.Slider
        SocialDLamp                     matlab.ui.control.Lamp
        QuarantineSwitch                matlab.ui.control.Switch
        QuarantineLabel                 matlab.ui.control.Label
        QuarantineSlider                matlab.ui.control.Slider
        QuarantineLamp                  matlab.ui.control.Lamp
        LockdownSwitch                  matlab.ui.control.Switch
        LockdownLabel_5                 matlab.ui.control.Label
        LockdownSlider                  matlab.ui.control.Slider
        LockdownLamp                    matlab.ui.control.Lamp
        FatalityRateLabel               matlab.ui.control.Label
        FatalitySlider                  matlab.ui.control.Slider
        DurationofInfectionLabel        matlab.ui.control.Label
```

```
        DinfSlider                      matlab.ui.control.Slider
        DurationofIncubationLabel       matlab.ui.control.Label
        DincSlider                      matlab.ui.control.Slider
        VirusDropDownLabel_2            matlab.ui.control.Label
        CountryDropDown_2Label_2        matlab.ui.control.Label
        BasicReproductionNumberLabel    matlab.ui.control.Label
        BRNSlider                       matlab.ui.control.Slider
        DincSwitch                      matlab.ui.control.Switch
        DinfSwitch                      matlab.ui.control.Switch
        FatalitySwitch                  matlab.ui.control.Switch
        BRNSwitch                       matlab.ui.control.Switch
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Callback function: BRNSlider, BRNSwitch, CountryDropDown,
        % DincSlider, DincSwitch, DinfSlider, DinfSwitch,
        % FatalitySlider, FatalitySwitch, LockdownSlider,
        % LockdownSwitch, QuarantineSlider, QuarantineSwitch,
        % SocialDSlider, SocialDSwitch, UpdateButton, VirusDropDown
        function UpdateButtonPushed(app, event)

        %% GLOBAL VARIABLES

        global N;
        global R_0;
        global death;
        global T_inc;
        global T_inf;

        global hosp;

        global beta;
        global gamma;
        global sigma;


```

```
68          %% PARAMETERS W.R.T DROP DOWN MENU SELECTIONS (LIBRARY
               SECTION)
69          % The parameters (or constants) of the equations to be used
               will have different values for a given region and virus.
70
71          switch char(app.CountryDropDown.Value)
72              case char('India')
73                  N=1.353e9;
74                  hosp=(N/1000)*0.53;
75                  switch char(app.VirusDropDown.Value)
76                      case char('SARS COVID-2 (2020)')
77                          R_0=2.9;
78                          death=1-0.9851;
79                          T_inc=4.2;
80                          T_inf=14;
81                      case char('Ebola (2014)')
82                          R_0=1.8;
83                          death=1-0.5;
84                          T_inc=9;
85                          T_inf=20;
86                  end
87
88              case char('Russia')
89                  N=144.5e6;
90                  hosp=(N/1000)*8.05;
91                  switch char(app.VirusDropDown.Value)
92                      case char('SARS COVID-2 (2020)')
93                          R_0=2.2;
94                          death=1-0.9828;
95                          T_inc=4.2;
96                          T_inf=14;
97                      case char('Ebola (2014)')
98                          R_0=1.8;
99                          death=1-0.5;
100                         T_inc=9;
101                         T_inf=20;
102                 end
103
```

```
104             case char('United Arab Emirates')
105                 N=9.631e6;
106                 hosp=(N/1000)*1.4;
107                 switch char(app.VirusDropDown.Value)
108                     case char('SARS COVID-2 (2020)')
109                         R_0=2.8;
110                         death=1-0.9963;
111                         T_inc=4.2;
112                         T_inf=14;
113                     case char('Ebola (2014)')
114                         R_0=1.8;
115                         death=1-0.5;
116                         T_inc=9;
117                         T_inf=20;
118                 end
119
120             case char('United States of America')
121                 N=328.2e6;
122                 hosp=(N/1000)*2.77;
123                 switch char(app.VirusDropDown.Value)
124                     case char('SARS COVID-2 (2020)')
125                         R_0=2.2;
126                         death=1-0.96;
127                         T_inc=4.2;
128                         T_inf=14;
129                     case char('Ebola (2014)')
130                         R_0=1.8;
131                         death=1-0.5;
132                         T_inc=9;
133                         T_inf=20;
134                 end
135
136         end
137
138
139         %% FUNCTIONALITY OF SWITCHES
140         % If the switch is off for a particular quantity, its value
              will be 0, else its value will take the default value
```

```
            assigned above.
141
142         if isequal(app.BRNSwitch.Value,'On')
143             R_0=app.BRNSlider.Value;
144         else
145             app.BRNSlider.Value=R_0;
146         end
147         if isequal(app.DincSwitch.Value,'On')
148             T_inc=app.DincSlider.Value;
149         else
150             app.DincSlider.Value=T_inc;
151         end
152         if isequal(app.DinfSwitch.Value,'On')
153             T_inf=app.DinfSlider.Value;
154         else
155             app.DinfSlider.Value=T_inf;
156         end
157         if isequal(app.FatalitySwitch.Value,'On')
158             death=app.FatalitySlider.Value;
159         else
160             app.FatalitySlider.Value=death;
161         end
162
163
164         if isequal(app.LockdownSwitch.Value,'Off')
165             app.LockdownSlider.Value=0;
166             app.LockdownLamp.Color=[0.90,0.90,0.90];
167         else
168             app.LockdownLamp.Color=[0,1,0];
169         end
170         if isequal(app.SocialDSwitch.Value,'Off')
171             app.SocialDSlider.Value=0;
172             app.SocialDLamp.Color=[0.90,0.90,0.90];
173         else
174             app.SocialDLamp.Color=[0,1,0];
175         end
176         if isequal(app.QuarantineSwitch.Value,'Off')
177             app.QuarantineSlider.Value=0;
```

```
178             app.QuarantineLamp.Color=[0.90,0.90,0.90];
179         else
180             app.QuarantineLamp.Color=[0,1,0];
181         end
182
183
184         %% FORMULAS
185
186         % BASIC
187         beta=R_0/(N*T_inc);
188         gamma = 1/T_inf;
189         sigma = 1/T_inc;
190
191         % ODE
192         t = 0:1:365;
193         y0 = [N-28, 0, 25, 3, 0];
194         [t,y]=ode45(@(t,y) ode_solve(t,y), t, y0);
195
196
197         %% PLOTTING
198
199         plot(app.UIAxesCalculated,t,y,t,100*hosp,'o','LineWidth', 2)
200         legend(app.UIAxesCalculated,'Susceptible','Exposed','
               Infectious','Recovered','Death','100*Capacity','Location',
               'Best')
201
202
203         %% ODE SOLVER FUNCTION
204
205         function dydt = ode_solve(t,y)
206
207         S = y(1);
208         E= y(2);
209         I = y(3);
210
211         dS = -(beta).*(1-app.QuarantineSlider.Value/3).*(1-app.
               SocialDSlider.Value/5).*I.*S - (app.LockdownSlider.Value
               /200).*S;
```

```
212         dL = (app.LockdownSlider.Value/200).*S;
213         dE = (beta).*(1-app.QuarantineSlider.Value/3).*(1-app.
                SocialDSlider.Value/5).*I.*S - sigma.*E;
214         dQ = (app.QuarantineSlider.Value/3).*I - (1-app.
                QuarantineSlider.Value/3).*gamma*(1-death)*I - (app.
                QuarantineSlider.Value/3).*(death)*gamma*I;
215         dI = sigma*E - (1-app.QuarantineSlider.Value/3).*gamma.*I - (
                app.QuarantineSlider.Value/3).*gamma.*I;
216         dR = gamma*(1-death)*I;
217         dD = gamma*(death)*I;
218
219         dydt = [dS; dE; dI; dR; dD];
220         end
221         end
222
223         % Button pushed function: DefaultButton
224         function DefaultButtonPushed(app, event)
225
226             %% DEFAULT SETTING
227             % Puts all the switches in 'off' configuration and that
                    triggers the update button which causes all the
                    sliders to take default value as in library section.
228
229             app.BRNSwitch.Value='Off';
230             app.DincSwitch.Value='Off';
231             app.DinfSwitch.Value='Off';
232             app.FatalitySwitch.Value='Off';
233
234         end
235     end
236
237     % Component initialization
238     methods (Access = private)
239
240         % Create UIFigure and components
241         function createComponents(app)
242
```

```
243             % Create UIFigure and hide until all components are
                    created
244             app.UIFigure = uifigure('Visible', 'off');
245             app.UIFigure.Position = [100 100 1363 869];
246             app.UIFigure.Name = 'MATLAB App';
247
248             % Create UIAxesCalculated
249             app.UIAxesCalculated = uiaxes(app.UIFigure);
250             title(app.UIAxesCalculated, '')
251             xlabel(app.UIAxesCalculated, 'Time Axis (Days)')
252             ylabel(app.UIAxesCalculated, 'SEIRD')
253             app.UIAxesCalculated.PlotBoxAspectRatio =
                    [1.66614420062696 1 1];
254             app.UIAxesCalculated.MinorGridLineStyle = '-';
255             app.UIAxesCalculated.XGrid = 'on';
256             app.UIAxesCalculated.XMinorGrid = 'on';
257             app.UIAxesCalculated.YGrid = 'on';
258             app.UIAxesCalculated.YMinorGrid = 'on';
259             app.UIAxesCalculated.Position = [13 9 941 577];
260
261             % Create SEIRDLQModelofContagiousVirusesLabel
262             app.SEIRDLQModelofContagiousVirusesLabel = uilabel(app.
                    UIFigure);
263             app.SEIRDLQModelofContagiousVirusesLabel.FontName = '
                    Arial Nova Light';
264             app.SEIRDLQModelofContagiousVirusesLabel.FontSize = 30;
265             app.SEIRDLQModelofContagiousVirusesLabel.FontWeight = '
                    bold';
266             app.SEIRDLQModelofContagiousVirusesLabel.Position = [531
                    776 511 59];
267             app.SEIRDLQModelofContagiousVirusesLabel.Text = 'SEIRD-LQ
                     Model of Contagious Viruses';
268
269             % Create CountryDropDown
270             app.CountryDropDown = uidropdown(app.UIFigure);
271             app.CountryDropDown.Items = {'India', 'Russia', 'United
                    Arab Emirates', 'United States of America'};
```

```
272             app.CountryDropDown.ValueChangedFcn = createCallbackFcn(
                    app, @UpdateButtonPushed, true);
273             app.CountryDropDown.FontSize = 16;
274             app.CountryDropDown.Position = [507 597 251 22];
275             app.CountryDropDown.Value = 'India';
276 
277             % Create VirusDropDown
278             app.VirusDropDown = uidropdown(app.UIFigure);
279             app.VirusDropDown.Items = {'SARS COVID-2 (2020)', 'Ebola
                    (2014)'};
280             app.VirusDropDown.ValueChangedFcn = createCallbackFcn(app
                    , @UpdateButtonPushed, true);
281             app.VirusDropDown.FontSize = 16;
282             app.VirusDropDown.Position = [239 597 251 22];
283             app.VirusDropDown.Value = 'SARS COVID-2 (2020)';
284 
285             % Create UpdateButton
286             app.UpdateButton = uibutton(app.UIFigure, 'push');
287             app.UpdateButton.ButtonPushedFcn = createCallbackFcn(app,
                     @UpdateButtonPushed, true);
288             app.UpdateButton.Position = [818 597 100 22];
289             app.UpdateButton.Text = 'Update';
290 
291             % Create DefaultButton
292             app.DefaultButton = uibutton(app.UIFigure, 'push');
293             app.DefaultButton.ButtonPushedFcn = createCallbackFcn(app
                    , @DefaultButtonPushed, true);
294             app.DefaultButton.Position = [82 597 100 22];
295             app.DefaultButton.Text = 'Default';
296 
297             % Create AyanAhmad2K19EP027DhruvTyagi2K19EP032Label
298             app.AyanAhmad2K19EP027DhruvTyagi2K19EP032Label = uilabel(
                    app.UIFigure);
299             app.AyanAhmad2K19EP027DhruvTyagi2K19EP032Label.FontSize =
                     16;
300             app.AyanAhmad2K19EP027DhruvTyagi2K19EP032Label.Position =
                     [576 707 422 24];
```

```
301             app.AyanAhmad2K19EP027DhruvTyagi2K19EP032Label.Text = '
                    Ayan Ahmad (2K19/EP/027) & Dhruv Tyagi (2K19/EP/032)';
302
303             % Create Image
304             app.Image = uiimage(app.UIFigure);
305             app.Image.Position = [360 692 144 143];
306             app.Image.ImageSource = 'DTU.png';
307
308             % Create
                    DepartmentofAppliedPhysicsDelhiTechnologicalUniversityLabel
309             app.
                    DepartmentofAppliedPhysicsDelhiTechnologicalUniversityLabel
                     = uilabel(app.UIFigure);
310             app.
                    DepartmentofAppliedPhysicsDelhiTechnologicalUniversityLabel
                    .FontSize = 18;
311             app.
                    DepartmentofAppliedPhysicsDelhiTechnologicalUniversityLabel
                    .Position = [536 747 503 30];
312             app.
                    DepartmentofAppliedPhysicsDelhiTechnologicalUniversityLabel
                    .Text = 'Department of Applied Physics, Delhi
                    Technological University';
313
314             % Create SocialDSwitch
315             app.SocialDSwitch = uiswitch(app.UIFigure, 'slider');
316             app.SocialDSwitch.ValueChangedFcn = createCallbackFcn(app
                    , @UpdateButtonPushed, true);
317             app.SocialDSwitch.Position = [1261 78 45 20];
318
319             % Create SocialDistancingLabel
320             app.SocialDistancingLabel = uilabel(app.UIFigure);
321             app.SocialDistancingLabel.FontSize = 16;
322             app.SocialDistancingLabel.Position = [999 77 129 22];
323             app.SocialDistancingLabel.Text = 'Social Distancing';
324
325             % Create SocialDSlider
```

```
            app.SocialDSlider = uislider(app.UIFigure);
            app.SocialDSlider.Limits = [0 0.9];
            app.SocialDSlider.ValueChangedFcn = createCallbackFcn(app
               , @UpdateButtonPushed, true);
            app.SocialDSlider.Position = [999 59 326 3];

            % Create SocialDLamp
            app.SocialDLamp = uilamp(app.UIFigure);
            app.SocialDLamp.Position = [969 50 20 20];
            app.SocialDLamp.Color = [0.902 0.902 0.902];

            % Create QuarantineSwitch
            app.QuarantineSwitch = uiswitch(app.UIFigure, 'slider');
            app.QuarantineSwitch.ValueChangedFcn = createCallbackFcn(
               app, @UpdateButtonPushed, true);
            app.QuarantineSwitch.Position = [1261 170 45 20];

            % Create QuarantineLabel
            app.QuarantineLabel = uilabel(app.UIFigure);
            app.QuarantineLabel.FontSize = 16;
            app.QuarantineLabel.Position = [999 169 84 22];
            app.QuarantineLabel.Text = 'Quarantine';

            % Create QuarantineSlider
            app.QuarantineSlider = uislider(app.UIFigure);
            app.QuarantineSlider.Limits = [0 0.9];
            app.QuarantineSlider.ValueChangedFcn = createCallbackFcn(
               app, @UpdateButtonPushed, true);
            app.QuarantineSlider.Position = [999 151 326 3];

            % Create QuarantineLamp
            app.QuarantineLamp = uilamp(app.UIFigure);
            app.QuarantineLamp.Position = [969 142 20 20];
            app.QuarantineLamp.Color = [0.902 0.902 0.902];

            % Create LockdownSwitch
            app.LockdownSwitch = uiswitch(app.UIFigure, 'slider');
```

```
360             app.LockdownSwitch.ValueChangedFcn = createCallbackFcn(
                   app, @UpdateButtonPushed, true);
361             app.LockdownSwitch.Position = [1261 261 45 20];
362
363             % Create LockdownLabel_5
364             app.LockdownLabel_5 = uilabel(app.UIFigure);
365             app.LockdownLabel_5.FontSize = 16;
366             app.LockdownLabel_5.Position = [999 260 78 22];
367             app.LockdownLabel_5.Text = 'Lockdown';
368
369             % Create LockdownSlider
370             app.LockdownSlider = uislider(app.UIFigure);
371             app.LockdownSlider.Limits = [0 0.9];
372             app.LockdownSlider.ValueChangedFcn = createCallbackFcn(
                   app, @UpdateButtonPushed, true);
373             app.LockdownSlider.Position = [999 242 326 3];
374
375             % Create LockdownLamp
376             app.LockdownLamp = uilamp(app.UIFigure);
377             app.LockdownLamp.Position = [969 233 20 20];
378             app.LockdownLamp.Color = [0.902 0.902 0.902];
379
380             % Create FatalityRateLabel
381             app.FatalityRateLabel = uilabel(app.UIFigure);
382             app.FatalityRateLabel.FontSize = 16;
383             app.FatalityRateLabel.Position = [999 352 95 22];
384             app.FatalityRateLabel.Text = 'Fatality Rate';
385
386             % Create FatalitySlider
387             app.FatalitySlider = uislider(app.UIFigure);
388             app.FatalitySlider.Limits = [0 1];
389             app.FatalitySlider.ValueChangedFcn = createCallbackFcn(
                   app, @UpdateButtonPushed, true);
390             app.FatalitySlider.Position = [999 334 326 3];
391
392             % Create DurationofInfectionLabel
393             app.DurationofInfectionLabel = uilabel(app.UIFigure);
394             app.DurationofInfectionLabel.FontSize = 16;
```

```
395             app.DurationofInfectionLabel.Position = [999 444 149 22];
396             app.DurationofInfectionLabel.Text = 'Duration of
                    Infection';
397
398             % Create DinfSlider
399             app.DinfSlider = uislider(app.UIFigure);
400             app.DinfSlider.Limits = [0 30];
401             app.DinfSlider.ValueChangedFcn = createCallbackFcn(app, @
                    UpdateButtonPushed, true);
402             app.DinfSlider.Position = [999 426 326 3];
403
404             % Create DurationofIncubationLabel
405             app.DurationofIncubationLabel = uilabel(app.UIFigure);
406             app.DurationofIncubationLabel.FontSize = 16;
407             app.DurationofIncubationLabel.Position = [999 535 162
                    22];
408             app.DurationofIncubationLabel.Text = 'Duration of
                    Incubation';
409
410             % Create DincSlider
411             app.DincSlider = uislider(app.UIFigure);
412             app.DincSlider.Limits = [0 30];
413             app.DincSlider.ValueChangedFcn = createCallbackFcn(app, @
                    UpdateButtonPushed, true);
414             app.DincSlider.Position = [999 517 326 3];
415
416             % Create VirusDropDownLabel_2
417             app.VirusDropDownLabel_2 = uilabel(app.UIFigure);
418             app.VirusDropDownLabel_2.HorizontalAlignment = 'right';
419             app.VirusDropDownLabel_2.FontSize = 16;
420             app.VirusDropDownLabel_2.Position = [345 635 40 22];
421             app.VirusDropDownLabel_2.Text = 'Virus';
422
423             % Create CountryDropDown_2Label_2
424             app.CountryDropDown_2Label_2 = uilabel(app.UIFigure);
425             app.CountryDropDown_2Label_2.HorizontalAlignment = 'right
                    ';
426             app.CountryDropDown_2Label_2.FontSize = 16;
```

```
427             app.CountryDropDown_2Label_2.Position = [604 635 58 22];
428             app.CountryDropDown_2Label_2.Text = 'Country';
429
430             % Create BasicReproductionNumberLabel
431             app.BasicReproductionNumberLabel = uilabel(app.UIFigure);
432             app.BasicReproductionNumberLabel.FontSize = 16;
433             app.BasicReproductionNumberLabel.Position = [999 625 207
                    22];
434             app.BasicReproductionNumberLabel.Text = 'Basic
                    Reproduction Number';
435
436             % Create BRNSlider
437             app.BRNSlider = uislider(app.UIFigure);
438             app.BRNSlider.Limits = [0 5];
439             app.BRNSlider.ValueChangedFcn = createCallbackFcn(app, @
                    UpdateButtonPushed, true);
440             app.BRNSlider.Position = [999 607 326 3];
441
442             % Create DincSwitch
443             app.DincSwitch = uiswitch(app.UIFigure, 'slider');
444             app.DincSwitch.ValueChangedFcn = createCallbackFcn(app, @
                    UpdateButtonPushed, true);
445             app.DincSwitch.Position = [1261 536 45 20];
446
447             % Create DinfSwitch
448             app.DinfSwitch = uiswitch(app.UIFigure, 'slider');
449             app.DinfSwitch.ValueChangedFcn = createCallbackFcn(app, @
                    UpdateButtonPushed, true);
450             app.DinfSwitch.Position = [1261 445 45 20];
451
452             % Create FatalitySwitch
453             app.FatalitySwitch = uiswitch(app.UIFigure, 'slider');
454             app.FatalitySwitch.ValueChangedFcn = createCallbackFcn(
                    app, @UpdateButtonPushed, true);
455             app.FatalitySwitch.Position = [1262 353 45 20];
456
457             % Create BRNSwitch
458             app.BRNSwitch = uiswitch(app.UIFigure, 'slider');
```

```
459             app.BRNSwitch.ValueChangedFcn = createCallbackFcn(app, @
                    UpdateButtonPushed, true);
460             app.BRNSwitch.Position = [1263 626 45 20];
461 
462             % Show the figure after all components are created
463             app.UIFigure.Visible = 'on';
464         end
465     end
466 
467     % App creation and deletion
468     methods (Access = public)
469 
470         % Construct app
471         function app = SEIRD_LQ
472 
473             % Create UIFigure and components
474             createComponents(app)
475 
476             % Register the app with App Designer
477             registerApp(app, app.UIFigure)
478 
479             if nargout == 0
480                 clear app
481             end
482         end
483 
484         % Code that executes before app deletion
485         function delete(app)
486 
487             % Delete UIFigure when app is deleted
488             delete(app.UIFigure)
489         end
490     end
491 end
```

References

Roy M Anderson, B Anderson, and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

Ivan Area, Hanan Batarfi, Jorge Losada, Juan J Nieto, Wafa Shammakh, and Ángela Torres. On a fractional order ebola epidemic model. *Advances in Difference Equations*, 2015(1):278, 2015.

Ivan Area, FaÏÇal NdaÏrou, Juan J Nieto, Cristiana J Silva, and Delfim FM Torres. Ebola model and optimal control with vaccination constraints. *arXiv preprint arXiv:1703.01368*, 2017.

Julien Arino and P Van den Driessche. A multi-city epidemic model. *Mathematical Population Studies*, 10(3):175–193, 2003.

Joan L Aron and Ira B Schwartz. Seasonality and period-doubling bifurcations in an epidemic model. *Journal of theoretical biology*, 110(4):665–679, 1984.

Norman TJ Bailey et al. *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

Ross Beckley, Cametria Weatherspoon, Michael Alexander, Marissa Chandler, Anthony Johnson, and Ghan S Bhatt. Modeling epidemics with differential equation, 2013.

Vincenzo Capasso and Gabriella Serio. A generalization of the kermack-mckendrick deterministic epidemic model. *Mathematical Biosciences*, 42(1-2):43–61, 1978.

David Fisman, Edwin Khoo, and Ashleigh Tuite. Early epidemic dynamics of the west african 2014 ebola outbreak: estimates derived with a simple two-parameter model. *PLoS currents*, 6, 2014.

Pau Fonseca i Casas, Víctor García i Carrasco, and Joan Garcia i Subirana. Seird covid-19 formal characterization and model comparison validation. *Applied Sciences*, 10(15):5162, 2020.

Alison Gray, David Greenhalgh, Liangjian Hu, Xuerong Mao, and Jiafeng Pan. A stochastic differential equation sis epidemic model. *SIAM Journal on Applied Mathematics*, 71(3):876–902, 2011.

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.

Herbert W Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

Olga Krylova and David JD Earn. Effects of the infectious period distribution on predicted transitions in childhood disease dynamics. *Journal of The Royal Society Interface*, 10(84):20130098, 2013.

Phenyo E Lekone and Bärbel F Finkenstädt. Statistical inference in a stochastic epidemic seir model with control intervention: Ebola as a case study. *Biometrics*, 62(4):1170–1177, 2006.

Alessandro Rizzo, Biagio Pedalino, and Maurizio Porfiri. A network model for ebola spreading. *Journal of theoretical biology*, 394:212–222, 2016.

Shigui Ruan and Wendi Wang. Dynamical behavior of an epidemic model with a nonlinear incidence rate. *Journal of Differential Equations*, 188(1):135–163, 2003.

Boris Shulgin, Lewi Stone, and Zvia Agur. Pulse vaccination strategy in the sir epidemic model. *Bulletin of mathematical biology*, 60(6):1123–1148, 1998.

Rama S Singh and Marcy K Uyenoyama. *The evolution of population biology*. Cambridge University Press, 2004.

Donald Worster. *Nature's economy: a history of ecological ideas*. Cambridge University Press, 1994.

Wuyue Yang, Dongyan Zhang, Liangrong Peng, Changjing Zhuge, and Liu Hong. Rational evaluation of various epidemic models based on the covid-19 data of china. *arXiv preprint arXiv:2003.05666*, 2020.