SCHOOL OF COMPUTER SCIENCE AND INFORMATICS

CARDIFF UNIVERSITY

2021

CMT 403 - Computing Dissertation

Optimal Control of SEIR Epidemic Models

Author : Neetash S Pataria, C2078482

Supervisor: Frank C Langbein

Moderator: Matthew J W Morgan

Degree Programme: MSc Computing

ABSTRACT

This report documents the completed work to perform a problem analysis through mathematical modelling, here we built a model and simulation to help pinpoint the problem zones concerning the control of any SEIR disease, in this report the disease is COVID-19. The main aim was the creation of a successful, qualified, accurate and quantified system model using mathematical modelling, which is the trusted way of modelling, simulating and predicting the outburst of any disease and analysing proposed or in understanding new control methods.

While there are ample studies being conducted to end this global pandemic, this project offers an opportunity to help COVID-19 researchers, the NHS and the healthcare workers to look at the model, analyse different scenarios and situations and better understand how we can tackle existing or upcoming problems better.

During the making of the project, there have been weekly progress updates with my project supervisor Frank C Langbein, to demonstrate the model creation, results and discuss further plans. The modelling and analysis is done using Python and its various data science libraries.

Keywords:
Mathematical modelling, Control theory, COVID-19, Monte carlo simulations

ACKNOWLEDGEMENTS

We would like to express my deepest sense of gratitude to my supervisor Frank C Langbein for supporting me throughout the project, providing me valuable guidance, motivation and for his extreme cooperation to complete this project.

I appreciate all my colleagues whose direct and indirect contributions helped me a lot to complete this project. I would also like to thank all my teaching and non-teaching staff here at Cardiff University for cooperating with me and providing valuable advice which helped me in the completion of this work.

Finally, I would like to express my thanks to my family, without whose support I would have never been able to aspire for this level of education.

Ultimately I would like to shine some light on NHS UK for all the incredible efforts they have done and are doing to handle the current COVID-19 pandemic, as well as its other healthcare workers.

TABLE OF CONTENTS

Abstract 1
Acknowledgements 2

1. Introduction 5
   1.1. Sketching the problem 5
   1.2. Aim 5
   1.3. Scope of the project 6
   1.4. Intended audience 7
   1.5. Assumptions 7

2. Background 8
   2.1. Wider context 8
   2.2. Problem definition 8
   2.3. Background research 10
      2.3.1. Methodology selected 10
      2.3.2. Epidemic modelling 13
      2.3.3. Control strategies or Optimal control theory 20
      2.3.4. Monte Carlo simulations 22
      2.3.5. Related research 27

3. Software analysis 29

4. Requirements 32

5. Implementation 33
   5.1. Epidemic modelling 33
   5.2. Optimal control theory 35
   5.3. Monte Carlo simulations 36

6. Results and Evaluation 39

7. Conclusion and Future work 44
   7.1. Conclusion 44
   7.2. Future work 44

8. Reflection on learning 46

9. References 48

TABLE OF FIGURES

Fig 1: An Elemental depiction of the scientific method that shows how our conceptual models of the world are related to observations made within that real world.

Fig 2: A First-order view of mathematical modeling.

Fig 3: The SIR compartmental model.

Fig 4: The SEIR compartmental model.

Fig 5: The SEIRS compartmental model.

Fig 6: Probability distribution over the quadrant to estimate the value of pi.

Fig 7: Various trajectories of creation of dimer and monomers along with their average.

Fig 8: Basic workflow.

Fig 11: advance() function following the Forward Euler method.

Fig 10: Converting parameter into a function.

Fig 11: Using the GEKKO package.

Fig 12: Flowchart for Gillespie algorithm.

Fig 13: SIR Model.

Fig 14: SEIR model.

Fig 15: SEIRS model.

Fig 16: Strict lockdown measures.

Fig 17: No lockdown measures.

Fig 18: Moderate lockdown measures.

Fig 19: Monte carlo simulations.

INTRODUCTION

1.1 Sketching the problem

We as a civilization have always thrived to constantly improve, aiming to achieve the best. For centuries we've innovated in our ways of living to make the best out of our limited resources. Although, with the population increasing everyday, it gets harder to provide for every individual. And with healthcare being the epicenter; all of us depend on being healthy to be able to function to our fullest. This is what this project is based upon, optimising and analysing ways to make our healthcare systems better more specifically with COVID-19, explained more briefly in the scope.

We are still in a pandemic, there is no question about the fact that COVID-19 has taken us all with a storm. We still have not completely recovered from it. Humans are considered to be social beings, and thus must be studied with relation to their environment. It is more advantageous to us if, for eg: cases of COVID-19 can be prevented by imposing a lockdown, or by the right vaccination policies than by successfully treating the declared cases of the virus.

There is no question that the healthcare systems around the world work round the clock to help make this world a better place. The current operating healthcare system in the UK is the National Health Service, or the NHS. Over 9 million people in the UK have had the virus(Our world in data. 2020), which means every 7th person in the country has had the virus. With such risks and consequences, it is critically important to tackle the problem and aim to achieve the best solutions for the citizens of the country.

1.2 Aim

Considering the previous section, a model for preventing the rapid spread of COVID-19 is required as soon as possible to provide smart healthcare services. For any project to start with, we must first analyse the problem. And thus pinpointing the problem was the very first thing on my plate.
The goal of this project is to successfully create a compartmental model and simulate results. Then applying optimising control techniques to these models, and finally checking for the correctness of the models.

Objectives:

- Research and learn about compartmental modelling.
- Choose a suitable programming language/softwares with appropriate libraries.
- Research and create a SIR model, and its other variants.
- Apply control theories to the models, one of lockdown policies, vaccination policies or impact of new medicines.

- Check for its robustness with a stochastic model.

1.3 Scope of the project

Unlike other parts of the project like sketching the problem, the scope is a part which is constantly being updated in the sense that it is being identified and discovered along its process. This section concerns the ethical considerations of this project.

Primarily, the project was to find an optimal control method to the SEIR epidemic models. The project was initially broken into two sections, one for the modelling and the next for the optimal control. From the initial proposal on PATS, the general idea was to make an SEIR model and then apply control techniques to it. Given the fact I am no expert in the field of medical science, it was very interesting for someone like me to investigate the problem in hand. Also, without any prior knowledge, I would be less biased to what the experts have to say about the topic, and thus I could add a fresh brain into the problem.

I began with a revision course on calculus and mathematics, as without solid knowledge of these, there was no point in going forward with the project. And along the side, I narrowed my research with the recommendations from my technical advisor, Dr. Frank C Langbein, on topics regarding SIR modelling.

After my initial explorations into the background, and after initial planning and brainstorming. I split my focus into the SIR and SEIR models. The idea was to completely understand the models, and then eventually try and increase its complexity to get more realistic and accurate results. After I had my two models ready and setup, the next thing to do was to add in a few more parameters to the model and increase its complexity, thus came the SEIRS model. And finally to have an aspect of control governing the model, this could be either lockdown methods, new variants or new medicines. With the pandemic still going on, the opportunity and the need to tackle this problem and to start building a model was very clearly evident. This I believed was a very realistic scope for the project.

In the later stage of the project, in discussions with my Technical advisor. He suggested I also add on an aspect to check the correctness of the model. This would show that the research that was done was correctly implemented and that the models were robust. Moreover my knowledge in electrical engineering has led me to investigate stochastic methods to do the same.

And Finally, because of the nature of this research project, it is very important to address the ethical considerations. The COVID-19 pandemic is a very sensitive topic and I needed to ensure I was being ethically correct and was working rightfully with the information available to me.

1.4 Intended audience

Once the project comes to a successful end, it would primarily help the NHS or any healthcare official who is looking to understand optimising a SIR (or its extended) model. The secondary beneficiary would be people who are more prone to the disease due to underlying or existing respiratory diseases. And to everyone in the country considering the domino effect, as healthcare officials make better decisions using these models, the citizens can expect the pandemic to end earlier with less number of people getting affected by it. And finally, countries can learn from one another to tackle such problems, making this a global domino.

1.5 Assumptions

The research proposed in the project is based on the assumptions and data presented online, in the form of research papers, university lectures or ebooks. I have assumed the data available online to be both correct and incorrect, and therefore gathered information only from sources I believe are reliable, or in some cases if the information is frequently repeated. This is crucial as for a project such as this, information keeps changing every minute and something which proved to be true earlier becomes inapplicable.

BACKGROUND

2.1 Wider context

The outbreak of any infectious disease can cause severe threats to mankind. With the ongoing spread of the novel coronavirus or COVID-19, we are now in the midst of an urgent global challenge to fight back. Taking a step back and looking at the bigger picture, we as a civilization are constantly moulding ourselves, modernizing and evolving at a pace unimaginable. Nevertheless, every now and then we are reminded of what is in our control and what is not. It made us realize health by far is our greatest asset without which we remain paralized.

With something like the COVID-19, which hit us like a storm. Being completely unexpected, we panicked, the healthcare workers around the globe had no idea on how to tackle such a situation. And with such rapid growth of the virus, researchers and scientists were forced to urgently produce results and develop models which can help us tackle or pause this outbreak.

It is no question about how important a lockdown is when a disease spreads linearly. But with this fact comes a very important question, when should we impose a lockdown? With millions of people already being victims to heart and lung diseases like Asthma, Bronchitis or even the common cold. They became a lot more vulnerable to COVID-19 than others. About 65 million people suffer from chronic obstructive pulmonary disease (COPD) and 3 million die from it each year, making it the third leading cause of death worldwide. About 334 million people suffer from asthma, the most common chronic disease of childhood affecting 14% of all children globally.(WHO - global impact of respiratory diseases)
There are several models on the covid 19 pandemic, yet not all are accurate or optimised. There needs to be a model which can help us make the right decision and make smart estimates of how to control this chaos. Also known in the medical industry as 'smart-healthcare'.

2.2 Problem Definition

From understanding the problem as a whole, let me change the direction of the problem, and dive in deeper and define the problem I am approaching.

Digging in deep, there are a lot of research papers and solutions online as to how to model an epidemic mathematically and solve it as a set of differential equations. But there are only a few which have a factor of control, to try and optimize the model as a whole. Having a factor of control, healthcare workers can now look at the model and make better decisions on when to implement a lockdown and what will be the most efficient way in doing so.

A number of questions would arise here and in order to understand the complexity of the problem and map out the entire scenario. Before solving any problem, it is very important and I believe crucial to get to the root causes of the problem and every factor which is related to it. I used the CATWOE analysis to do so, CATWOE analysis is used to analytically solve any problem, or simply it is a checklist for solving problems. Each letter in "CATWOE", defines a key element to problem-solving(as shown in Table1: CATWOE Analysis).

Root Definition Element	**Root Definition Description**	**Identified Definition**
C - Customers	Who are the beneficiaries of the research and how the issue affects them?	Primary: COVID-19 patients, people with other lung-related problems, NHS staff Secondary: NHS management, UK government
A - Actors	Who is involved in the situation?	NHS UK and all people in the country.
T - Transformation	How will the change I want will be affected in practice?	Identify an optimal solution to help the NHS make better decisions for the future.
W - World View	The 'purpose', what this research wants to achieve?	The healthcare system everywhere around the globe was struggling with managing the amounts of patients and making better decisions for our wellbeing. The focus of this research is on the United Kingdom however it could be used as a general-purpose model making use of the right data respectively.
O - Owners	Who controls the system and is accountable for it?	The NHS UK, and the entire population. As the NHS can terminate all its operations while the population can stop using the NHS.
E - Environment constraints	The factors that influence how the system works? Could be political, social, economic or natural.	The education system, TV and media, businesses, natural disasters, social pressures(habits, or

		consumption)

Table 1: CATWOE Analysis

This helped me analyse my problem better and allowed me to carry on my research in a stream of directions.

2.3 Background research

2.3.1 Methodology selected

The project was proposed by my supervisor Frank C Langbein, after discussions with him regarding the project. The project became of great interest to me, due to a few reasons. Firstly, this solves one of the issues that we are going through right now. Secondly, with my background as an electrical engineer, this project gave me the perfect opportunity to put all my leanings into one. My knowledge of control theory, mathematical modelling and calculus were all put to the test along with its application all using computer science principles. And with the technical advice from Frank C Langbein, I kept track to achieve this.

First and foremost, the introduction and definition of mathematical modeling are key. Beginning this by defining the simple term model:

model *(n): a miniature representation of something; a pattern of something to be made; an example for imitation or emulation; a description or analogy used to help visualize something that cannot be directly observed; a system of postulates, data and inferences presented as a mathematical description of an entity or state of affairs*

This means modelling is basically describing how objects or things we're keen to know about behave. Mathematical modelling is thus the representation of the model in mathematical terms.

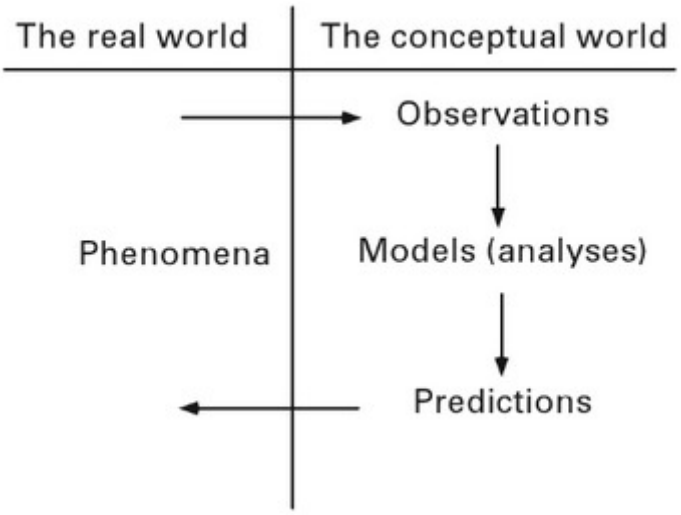


Fig 1: An Elemental depiction of the scientific method that shows how our conceptual models of the world are related to observations made within that real world(Dym and Ivey, 1980)

Thanks to high-speed computing power, mathematical modeling can help solve very complex problems. As Xavier Avula states in his book 'Encyclopedia of Physical Science and Technology', "Successful applications of mathematical modelling techniques in engineering sciences have led the way to extend the techniques to more exotic areas of inquiry."
Mathematical modelling, in practice it has both principles we need to apply, and the methods we are to use in order to model something. Below is the most basic algorithm on how we can mathematically model a system.

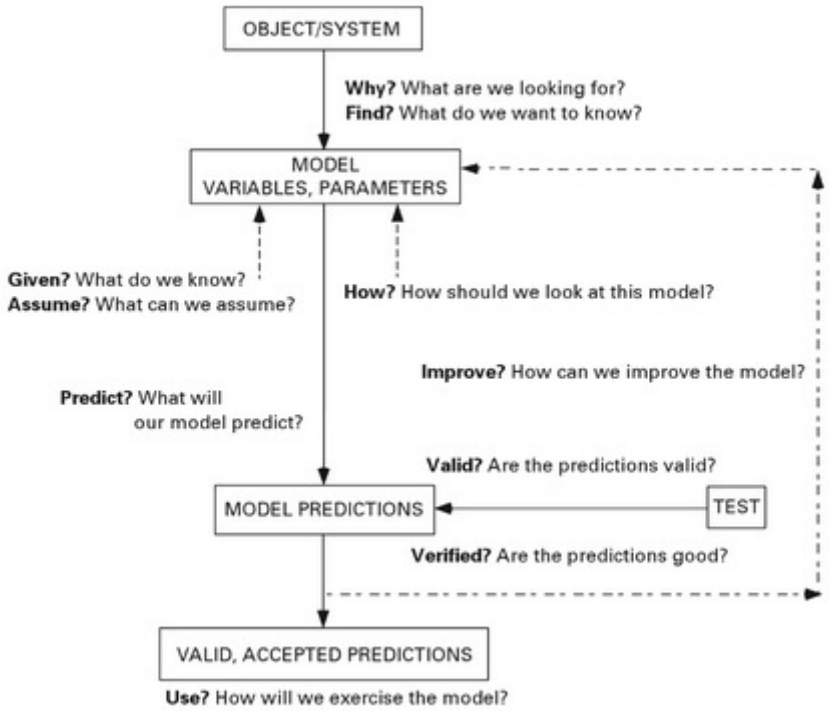


Fig 2: A First-order view of mathematical modeling(Taken from "Principles of mathematical modeling, Clive L. Dym")

On the other end of the spectrum, an alternate methodology could be agent-based modeling, which essentially is a bottom-up approach, where the model simulated the actions of an 'agent', individuals or group of individuals or an organization, all to understand the system behavior. While mathematical modelling is a top-down approach. This is the main difference between the two, where mathematical system modeling concerns the behavior of the system while agent-based modeling is that of an individual.

Mainly, for this reason, the chosen methodology is modelling the epidemic system using mathematical modelling as it focuses on the complete behaviour of the system and applications of relevant real-world complex systems, which fits the requirements of this research perfectly. Mathematical modelling enables us to create complicated models of the system and later apply control theories to it. Agent based modelling on the other hand does

not. Moreover, this will lead me to understand the problem with the help of the CATWOE analysis method (see section 6.2 *Problem definition*), developed for soft system modelling(SSM) by Peter Checkland.

Following the above step, the approach requires to begin the creation of mathematical models, in our project, the SIR Model, SEIR Model and the SEIRS Model (see section 6.3.2 *Epidemic modelling*), or the epidemic models in general. Throughout the creation of these models, it is vital to understand the core mathematics behind this. These models represent a set of ordinary differential equations which can be solved with the help of Python, or any mathematical programming language. These epidemiology models are very powerful and can completely change our current story, given the parameters are based on our assumptions and thus must be verified with justification. We can then run the simulations and observe the trends that any linearly spread disease such as COVID-19 follows. The parameters that go in as input can be played along with to view different scenarios in the model, the very first model is the most basic epidemic model of any disease that spreads linearly. Moving on we slowly increase the complexity of the model making it as realistic as possible. This means the goal is to have a model that can depict the current real life scenarios, and have the ability to integrate some control theory into the model. To be able to see what is best for everyone and how a change in one elemental parameter can change the shape of the model.

As mentioned above, after the model(s) is ready we then move to the control aspect of things. There can be a lot of scenarios, from vaccination policies, to lockdown measures to herd immunity or even future predictions with the development of new medications. The chosen control aspect for this research are the lockdown measures. The measures can be divided into three general categories, first is the worst case scenario i.e; no measures taken or no lockdown, secondly some form of measure or partial lockdown and finally a complete lockdown going into effect. Once these are understood and set, we then run simulations for the model. These will show us the trends which are going to help us guide our policy actions or interventions, or even recommendations to different proposed solutions. The chosen control measure is so because the other aspects would require some form of a machine learning algorithm in place to find the most optimal solution, which is out of the scope of this research project.

The final step in the methodology is to check for the correctness or robustness of the models that we have built. We do so with the help of Monte Carlo Simulations. As Will Kenton mentioned, “Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models”. Given the uncertainty in making the models with so little data and resources available, the Monte Carlo simulations are the perfect fit in the case for our research project. There are a lot of ways to simulate the Monte Carlo simulations, the one I used is the Gillespie Algorithm(see section 6.3.4 *Monte Carlo simulations*). The results

of these validate our previous simulations and results, showing our model is accurate and reliable.

2.3.2 Epidemic modelling

According to Daley and Gani (2005), John Gaunt was amongst the first who systematically, or mathematically tried to analyse and quantify deaths and its causes in his book *Natural and Political Observations made upon the Bills of Mortality (1662),* Daley and Gani also quoted this theory to be, "a theory that is now well established among modern epidemiologists". Epidemic modelling, in its exact definition, dates back to 1760. Daniel Bernouli, a Swiss mathematician and physicist, was one the first people to model effects of a disease for a fixed population. He designed a mathematical model to show inoculation of the smallpox virus, the results from his model showed that inoculation of the smallpox decreased life expectancy.
McKendrick in 1926 was the first scientist to include the factor of randomness in an epidemic model, where he observed real life outbreaks. This particular model is a stochastic model of the Deterministic General Epidemic Model. But it was only until 1949, when Bartlett observed the model by McKendrick extensively. And since then, all studies related to scientific modelling in the field of epidemiology have been directed towards statistical analysis.

This brings us to an important question, why use stochastic modelling in emerging diseases?
It is extremely crucial for the NHS or any healthcare official to make predictions concerning the rise of any new disease which causes infections or fatalities, to be able to make meaningful predictions about the future concerning the pandemic. And when a new disease is rising, there is not enough data or knowledge to base any such decisions. Moreover with no immunity in people concerning the pathogen, there is a risk of a very high outbreak. Stochastic modelling is one of the most reliable tools for making predictive decisions and assessing control strategies.

The randomness of these kinds of disease outbreaks, and its potential to spread throughout the globe rapidly via medium of air, these diseases are a global problem from the very first day. SARS (Severe Acute Respiratory Syndrome), Avian influenza and COVID-19 are a few examples in recent history. This type of a problem can be solved using the Stochastic model, it essentially is a tool to estimate probability distribution of the various possible outcomes with the random variable as input(s) with respect to time.

The SARS model has been the center of several studies in order to find the reproduction number and to understand the transmission dynamics in order to improve the intervention strategies. *Steven Riley*(2003), along with his team modelled data of the first 10 weeks of the SARS outbreak, they came to an estimate for the reproduction number of 2.7, i.e; 2.7 secondary infections were generated per human on an average in the first 10 weeks. There

were other research articles based on the above model and they found out the reproduction number to lie between 2.2 and 3.6. All these researches were based on deterministic modelling of the SARS virus. A deterministic model is used when we have a very large population, the population is divided into groups and assigned compartments (this type of modelling is also called compartmental modelling), where each compartment represents a certain stage in the epidemic.

In order for us to know control theories regarding this epidemic, it is essential for us to use the deterministic model. This way we can see the results of each control policy and how that is affecting each compartment in the model. This research is thus based on the SIR epidemic model, the SEIR and the SEIRS compartmental model.

SIR Model

The standard SIR model is the most basic of all the models, it is at the very base of all deterministic epidemic modelling.

In this model, the population in consideration is divided into three groups or compartments - namely Susceptible, Infected/or Infective and Recovered. There is a transition of people between these compartments. Susceptible are the people who can potentially have the virus, then we have the infected people who have confirmed to have the virus and can thus spread to other people. And finally we have the Recovered, these are the group of people who have either recovered from the virus or have died from it. The complete pandemic is too complicated to express as a set of ordinary differential equations, thus we make a number of assumptions (explained below) in order to simplify the model.

1. The epidemic does not last for a long time, in order for the population to stay constant (can also be vice versa).
2. The rate of increase in infectives is always proportional to contact between susceptible and infectives.
3. In the recovered compartment, the rate for death or recovery remains constant.

Fig 3:The SIR compartmental model

Let us consider the total population to be N, the number of susceptible people be S, the number of Infected people be I and the number of recovered people be R. The effective

contact rate is going to be β (1/min) i,e; the rate at which disease spreads in the population, and the recovery rate/mortality rate be γ (1/min).

$$S \xrightarrow{\beta I} I \xrightarrow{\gamma} R, \tag{1}$$

where ,

Total population: N = S + I + R
Susceptible: S = N - confirmed
Infected: I = confirmed - R - deaths
Recovered: R = recovered + deaths
Contact rate: β
Recovery rate: γ

As the number of recoveries or deaths increase, the number of people in the infected compartment decreases. While the people who come out of the recovered compartment, cannot go back to the susceptible compartment.

Following are the ordinary differential equations(ODES) for the SIR model:

$$\frac{dS}{dT} = -\frac{\beta SI}{N}$$

$$\frac{dI}{dT} = \frac{\beta SI}{N} - \gamma I$$

$$\frac{dR}{dT} = \gamma I$$

Where all the variables/constants are mentioned above, and T is the elapsed time beginning from the start of the pandemic. These ODEs are used for simulating the model. At any given point in time, these ODEs show the rate of change in every single variable.
Starting with the first equation, i.e; Susceptibles. The rate of change of the number of susceptible over time. From our second assumption, we know that the number of susceptibles are going to decrease over time as people move to the infected compartment. Thus we have the minus sign, with it decreasing at the rate of contact β. And again from the second assumption, it is proportional to the Infected and the Susceptible people. The SI, indicated some sort of a contact between the two compartments taking in consideration the total population N.

Coming to the second equation, i.e; Infected. The rate of change of the number of infected over time. Since the susceptibles are moving to the infected with the rate of change β, we have the same element from the previous equation, but now with a plus sign. Now from the

third assumption, we have the element of γI with a minus sign, signifying the number of infected people moving into the recovered compartment with the rate of γ.

Finally to the final equation, i.e; Recovered. The rate of change of the number of recovered over time. This is the gain from the previous equation keeping the third assumption in mind, but with a plus sign signifying people moving into the recovered compartment.

In order to solve the equations, we have to set the initial conditions for these equations. We do so by giving S an initial value of S_0, giving I an initial value of I_0. Finally we give R the value of 0 as we do not expect there to be any person who is recovered at the very start of the epidemic.
Finally, talking about the very first assumption in the context of the model (ODEs), it states the population must always remain constant. This means that the sum of the rate of change of the susceptible, infected and recovered come out to be zero, since N = S + I + R. All of these together make up our SIR model.

Now with our model ready, we can start asking questions.
Will the disease spread?
If the number of people in the infected compartment starts growing, then we have a spread. This makes the rate of contact β very important. We must thus study the second ODE, but before we do that we must first understand the first one. This is because it clearly states that the rate of change of susceptibles is negative, given the three elements in the equation are all positive values (β is a constant, I and S are people in a population and N is constant). Thus we can say that S is always going to be smaller than its initial value, theoretically it makes perfect sense, as initially everyone in the population is going to be susceptible in the population.

$$S \leq S0$$

If we consider, and plug this particular value of S into the second equation. We now have an inequality.

$$\frac{dI}{dT} < \frac{\beta S0 I}{N} - \gamma I$$

$$\frac{dI}{dT} < I(\frac{\beta S0}{N} - \gamma)$$

And we know that an epidemic will occur if I increases from I_0, or mathematically in the equation, the value of S_0 has to be less than γ/β. This value if flipped, i.e

$$\frac{1}{S0} = \frac{\beta}{\gamma}$$

This particular value is called the contact ratio. As defined by Dr. Tom Crawford at the University of Oxford said, “This is the fraction of the population that comes into contact with an infected individual during the period when they are infectious”.
We can also look at this ratio from a little different standpoint. By rearranging, multiplying with β and dividing by γ, we get a new parameter we call R_0.

$$R0 = \frac{\beta S0}{\gamma}$$

This is the basic reproduction number or the basic reproductive ratio. If this particular value is greater than 1, meaning I increases from I_0 and thus we will have an epidemic. Again defined by Dr. Tom Crawford as, “this number represents the number of secondary infections in the population caused by one initial primary infection”, simply put if a single person has the virus, then this number tells us how many other people this one infected person is going to average.
The exact number for this is still being determined for COVID-19, but it is estimated to be somewhere between 2.5-4 based on studies for particular geographical locations.

The complexity of the model can be increased by adding in another compartment called the Exposed compartment. This model is called the SEIR model.

SEIR Model

Increasing the complexity of the model, we now have the SEIR model.
In this model, the population in consideration is divided into four groups or compartments instead of three - namely Susceptible, Exposed, Infected/or Infective and Recovered. All three compartments taken from the SIR model serve the same purpose, the new edition is the Exposed compartment: These are the people who have acquired the disease/virus but are yet not contagious, or in simpler words there is a small period of time (called the latency period or the incubation period) between a person acquiring the virus and for the person to be able to spread it to others . This model, much like the SIR model, also has a number of assumptions (explained below) in order to simplify the model.

1. The epidemic does not last for a long time, in order for the population to stay constant (can also be vice versa).
2. The rate of increase in infectives is always proportional to contact between susceptible and infectives.
3. The latency period remains constant.
4. In the recovered compartment, the rate for death or recovery remains constant.

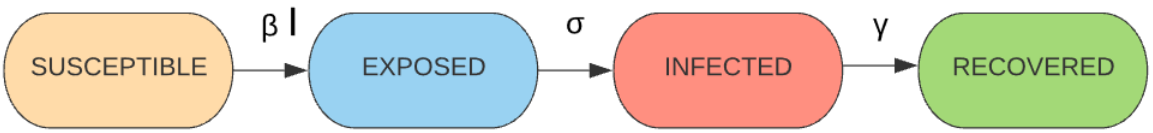


Fig 4: The SEIR compartmental model

We consider the same parameters as that of the SIR model, with an addition of σ(1/min) which is the latency period. The total population to be N, the number of susceptible people be S, the number of Infected people be I and the number of recovered people be R. The effective contact rate is going to be β (1/min) i,e; the rate at which disease spreads in the population, and the recovery rate/mortality rate be γ (1/min).

Following are the ordinary differential equations(ODES) for the SIER model:

$$\frac{dS}{dT} = -\frac{\beta SI}{N}$$

$$\frac{dE}{dT} = \frac{\beta SI}{N} - \sigma E$$

$$\frac{dI}{dT} = \sigma E - \gamma I$$

$$\frac{dR}{dT} = \gamma I$$

The way we write the differential equations are exactly the same as that for the SIR Model in the previous section.

SEIRS Model

Further increasing the complexity of the model, we can come to the SEIRS model.
In this model also, the population in consideration is divided into four groups or compartments exactly like the SEIR model- namely Susceptible, Exposed, Infected/or Infective and Recovered. The complexity is increased in the sense that the people who move out of the Recovered compartment can now come back to the Susceptible compartment. In simple terms we now have a cycle, only in a few cases complete this cycle, but the logic stands correct as we've seen with the current pandemic. Anyone who has tested positive for the virus, and recovered, can be exposed to the virus(or its variant) and be infected again thus making him/her susceptible even after recovery. To further add to the model, we can add the birth rate and the death rate of the population into account to see its effects on the

model, commonly known as demographics. These can also be eliminated if simulations are not feasible.

The assumptions made for the SEIRS model are same as that for the SEIR model with the addition:

1. For many respiratory infections, immunity after recovery is temporary and recovered individuals will lose immunity and return to S after an average protected period of $1/\omega$.
2. Demographics contribute to the in and out flows in individual groups based on the birth and death rate μ .

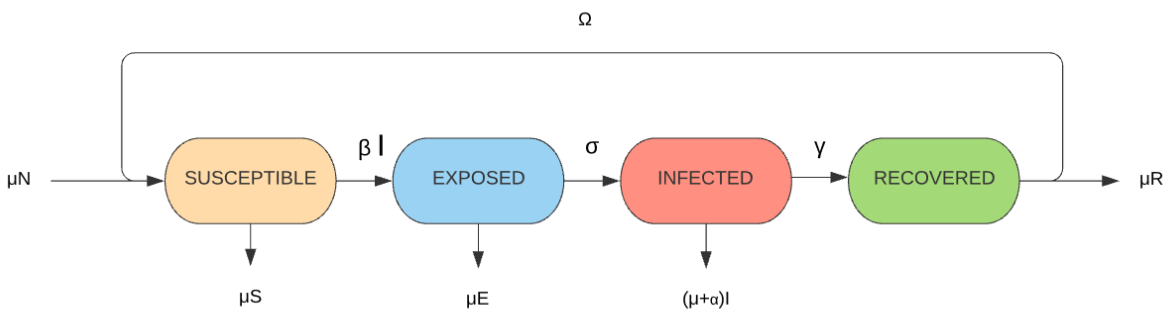


Fig 5: The SEIRS Model

The logic behind writing the differential equations are again exactly the same as we discussed in the previous sections.

The ordinary differential equations for the SEIRS model:

$$\frac{dS}{dT} = \mu N - \frac{\beta SI}{N} + \Omega R - \mu S$$

$$\frac{dE}{dT} = \frac{\beta SI}{N} - \sigma E - \mu E$$

$$\frac{dI}{dT} = \sigma E - \gamma I - (\mu + \alpha) I$$

$$\frac{dR}{dT} = \gamma I - \Omega R - \mu R$$

The paper 'The SEIR model for infectious disease dynamics' estimated the, "The associated basic reproduction number is

$$R_0 = [\sigma/(\sigma+\mu)] \times [\beta/(\gamma+\mu+\alpha)]$$

because the infectious period is $1/(\gamma+\mu+\alpha)$ and the probability of the index case becoming infectious rather than dying while in E is $\sigma/(\sigma+\mu)$. For most acute infections, μ is much smaller than the epidemic rates so realistic values do not appreciably alter the trajectories".

2.3.3 Control strategies or Optimal control theory

Control strategies or optimal control theory, deals with mathematical optimization where we find the control techniques for a system over a period of time so as to optimize a function. The idea behind these techniques is to develop an algorithm that will govern the inputs given to the system, and will drive the system to any desired state, with reducing any errors or delays or simply to get optimised results.
Optimal control is a mathematical optimization method in order to get control policies on a certain problem.

In this project, optimal control theory will be used in order to see the various lockdown measures that can be implemented based on the models we have previously made. There are other control policies that can also be implemented using the control theory such as Vaccination policies, or the introduction of a new variant, or the combination of two or more.

Going by the mathematics of optimal control, it might get a little overwhelming. So let us take a step back and understand with a simple example I thought of.
Consider sitting home on a hot summer day and thinking of having ice cream. Now one would think of going to the local supermarket and getting ice cream, while someone else might be ordering ice cream online. Both these steps have some advantages and drawbacks, one might argue it is too hot to go outside or deliveries may take a lot of time. While a control engineer might think of something completely different, he would consider all the possibilities and think of the most efficient way of getting the ice cream. Not going into too much detail, we need to find the quickest way to get the job done.
Bringing this logic to our problem, we need to find the fastest way to end the pandemic, and so what should be the lockdown measures that we need to have in place in order for us to call that the pandemic is officially over.

Mathematically, in the most simplest of statements:

We choose an action a(t) such that $s(t) = s^*(t)$, where s(t) is the current state while $s^*(t)$ is the desired state. This condition is often impossible to achieve as we cannot have the desired state exactly “now”.
So we write the action a(t) such that $s(t) \approx s^*(t)$, where we approximate the desired state to be the equal to the current state.
Quantifying this, we choose the action a(t) so that the chosen cost function(in our case reducing the length of the pandemic); C(s(t)) is minimal.

In order to get to the desired state, I used the GEKKO optimization suite(see *Section: Implementation*), and used iMode 7 which is Sequential simulation(SQS).

J. Jaime Gómez-Hernández and Eduardo F. Cassiraga in their paper "Theory and practice of sequential simulation", defined sequential simulation as, "Sequential simulation is a powerful stochastic simulation technique the theory of which relies on the ability to determine, for a given multivariate model, the conditional probability of a single random variable given any number of conditioning values. Sequential simulation can be used with those multivariate models for which these conditional probabilities can be determined. In practice, it is not enough to know how to determine the conditional probabilities, the procedure must be feasible from an operational point of view. Because it is not so in most cases, some approximations to the conditional probability distribution function are used in the implementation of the technique. These approximations are shown not to have a large impact in the performance of the technique, at least for the case in which the underlying multivariate model is multiGaussian."

In simpler terms, to bring this in the context of our problem, sequential simulation is a process of simulating the various pathways that a particular person will take. To simulate, a potential, infinite number of different scenarios and pathways. And finally present us with the most optimal one. It has a lot of applications in the field of medical science:

1. Help in designing and producing new approaches in the medical industry.
2. Testing for new interventions and changes.
3. Helping people understand present and future medical systems.

These are just a few applications, all of which are of paramount importance to the project. Specially points 1 and 2.

To understand how the simulation works, I took reference to another paper by J. Jaime Gómez-Hernández, 'One step at a time: The origins of sequential simulation and beyond', which very briefly explains sequential simulation, summarized as below:

Let us consider a completely random function $\{Z(u), u \in D\}$, characterized by all the n-variate distribution functions for any n-tuple of points within D.

$$F(Z(u_1), Z(u_2),...,Z(u_n)), \forall n | (u_1, u_2,...u_n) \in D$$

Now according to the definition above, we need to take a continuous variable and transform it into a collection of indicator variables.
Our random function, as defined above, can be converted or transformed into a set of indicator random functions, $\{I_k(u), k=1,...,K\}$, corresponding to the maximum value of K thresholds z_k given by:

$$I_k(u) = 1, \quad \text{if } Z(u) \leq z_k$$

Or

$$I_k(u) = 0, \quad \text{otherwise}$$

Here z(u) would be equal to $\{i_k, k=1,...,K\}$, which is being transformed to z(u).

From the earlier understanding of sequential simulation. Any multidimensional random function distribution can be rewritten as the product of univariate(observations on a single type of attribute) distributions by recursively using conditional probability on it.
Thus we can write:

$$\begin{aligned} F(Z(u_1), Z(u_2),\ldots,Z(u_n)) = {} & F(Z(u_1))\cdot \\ & F(Z(u_2)|Z(u_1))\cdot \\ & F(Z(u_3)|Z(u_1), Z(u_2))\cdots \\ & F(Z(u_n)|Z(u_1),\ldots,Z(u_n\text{-}1)) \end{aligned}$$

Where each component is the conditional distribution of a random variable given all the random variables that appear before the product.
The extension to generate realizations conditional to a set of (n_0) data is given by:

$$\begin{aligned} F(Z(u_1), Z(u_2),\ldots,Z(u_n)|(n_0)) = {} & F(Z(u_1)|(n_0))\cdot \\ & F(Z(u_2)|(n_0), Z(u_1))\cdot \\ & F(Z(u_3)|(n_0), Z(u_1), Z(u_2))\cdots \\ & F(Z(u_n)|(n_0), Z(u_1),\ldots,Z(u_n\text{-}1)) \end{aligned}$$

The simulation algorithm for the generation of a realization of a random function over a point set of size N conditional to n_0 observation data would be as follows:

1. Define a permutation of the numbers 1 to *N* that will identify the sequence in which the conditional univariate distributions will be built.
2. Sequentially visit all nodes according to the previous ordering and compute, at each node *i*, the conditional distribution of variable $Z(u_i)$ given the (n_0) data and all previously simulated random variables $\{Z(u_1)=z_1, Z(u_2)=z_2,\ldots,Z(u_{i-1})=z_{i-1}\}$

$$F(Z(u_i)|(n_0), Z(u_1)=z_1, Z(u_2)=z_2,\ldots,Z(u_{i-1})=z_{i-1})$$

3. Draw a value z_i from this distribution and incorporate it to the conditioning data set for the simulation of the next node.
4. Go back to step 2 until all nodes have been simulated.

2.3.4 Monte Carlo Simulations

These simulations got their name from the capital of Monaco, Monte Carlo. And with context to random sampling, this refers to the randomness in gambling which Monte Carlo is famous for. Monte Carlo simulations are a set of computing algorithms which apply repeated random sampling to get results. In simple words, these are simulations which over time

evolve randomly. These were developed by John von Neumann and Stanislaw Ulam during the second world war to help make better decisions during those uncertain times. Ever since, these have been used to assess risk in many scenarios ranging from AI, stock markets, sales and marketing, and of course epidemiology. These simulations have shown significant improvements over modelling, for eg; the ability to see the correlation between any two inputs. Or even sensitivity analysis, with the help of which we can see the impact caused by individual input parameters.

Researchers and engineers at IBM gave us a set of steps in order to use the Monte Carlo simulations given below(*IBM cloud education, 2020*):

1. Firstly set up the predictive model, identifying both the dependent variable to be predicted and the independent variables(also known as the input, risk or predictor variables) that will drive the prediction.
2. Specify probability distributions of the independent variables. Use historical data and/or the analyst's subjective judgement to define a range of likely values and assign probability weights for each.
3. Run simulations repeatedly, generating random values of the independent variables. Do this until enough results are gathered to make up a representative sample of the near infinite number of possible combinations.

We can now ask some important questions, are these methods accurate? And why do we need this?

Coming to the first question, checking the accuracy of these simulations. Let's take the famous example of finding the value of π.
Imagine a circle is drawn with its center at (0,0) and radius 1. Now we isolate a quarter of this circle and then inscribe it inside a square of side 1, same as that of the radius of the circle. We now select N number of points in the range (0,1), let the independent uniform distributions be X and Y. Or $X^2 + Y^2 < 1$.
Experimentally we found out that the proportion of the number of points falling inside the quadrant to the total number of points is π/4.
The standard deviation of the observed distributions is $\sqrt{\frac{(1-\pi/4)\pi/4}{N}} \approx \frac{0.41}{N}$. This implies that the higher the value of N, the more accurate the results are.
This said, risk engineers and analysts do not completely rely on this, the accuracy also depends on the precisions of the computations and the quality of the random number generator. Although there are many ways to calculate the value of π a lot faster and more stable, Monte Carlo simulation is the best when it comes to higher order differential equations.

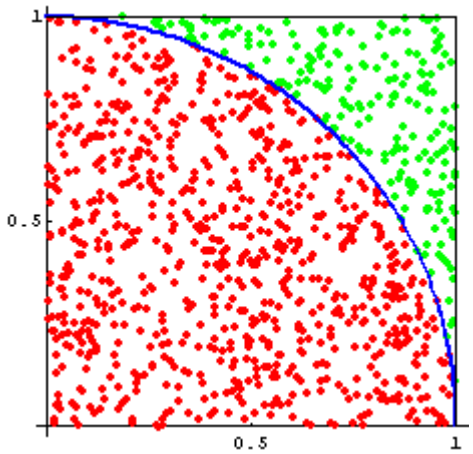


Fig 6: Probability distribution over the quadrant to estimate the value of pi.

We now know the accuracy of the Monte Carlo simulations can be controlled and thus we can make these really accurate. This answers our second question on why we need the Monte Carlo simulations.

I used the Monte Carlo simulations to check for the robustness of the system. Now that we know the Monte Carlo simulations are very accurate and trustworthy.
We can check if our models match with the monte carlo simulations, if so, it means that the calculations and predictions are correct and we can thus validate the model.

There are a lot of different ways in which we can carry out a Monte Carlo simulation. I choose to solve it with the help of the Gillespie Algorithm, also known as the SSA(Stochastic simulation algorithm) method.
The Gillespie algorithm is the original stochastic modelling method, this is because this was developed by a lot of people thus this algorithm evolved over time. But the name is accredited to the last person who took the last step in inventing it, Dan Gillespie. The method started off as to see the evolution of time for different stochastic processes, to exponentially distributed systems, and even to study epidemic outbreaks. Dan Gillespie wrote a paper on biochemical systems and solved them using the Gillespie algorithm, which then popularized the algorithm.

In context with mathematics and probability theory, the Gillespie algorithm creates a trajectory of the most possible solutions to a system of differential equations or any stochastic system whose reaction rates are known.
The Gillespie model simulates individual reaction events, this is done by employing two pseudorandom numbers. The first number chooses the time of the very next event, while the next number chooses the next reaction. The history of these events occurring are then essentially recorded, and given that we know that the numbers are chosen at random. Every event will come out differently, but if we repeat over and over again. We will begin to see a pattern, and as we increase the number of times we run the simulations. The results from the Gillespie algorithm will approximate the results obtained from the simulations of the

ordinary differential equations, or in some other scenarios these stochastic simulations may capture characteristics lost in the common modelling methods.

In order to implement any Gillespie Algorithm for any system, we start off by writing out all the possible events that could occur. Each event has a certain probability associated with it, and this affects the next event in the algorithm, or the choice of the next event is based on the probability of the previous one. This is the initial setup, and can be made specific to our system.
The next step is to choose two random numbers; First number decides what the next reaction will be, the selection is random but is weighted based on the individual rates of all the events in the system. While the second number decides when the event will occur.
After the event is chosen, the algorithm will adjust all the concentrations of the individual components to account for whatever event took place. This entire process of choosing the event and updating is iterated to create a timeline for a single chain of random events until some time(which will be set initially), has elapsed. The simulation is run from the starting conditions many times, ideally as many times as feasible.
With a large set of data, we can combine the data in a number of ways to represent the behaviour for the system.
This way of modelling will give us a more precise model compared to the standard ODE models initially created.

To summarize and simplify the above implementation, we can consider the steps by Lewis Cole, which are as follows:

1. **Initialization** - initialize the system, in the context of reaction kinetics this amounts to the setting up initial chemical concentrations
2. **Monte-Carlo** -
    1. Randomly simulate the time to the next event
    2. Given an event has occurred randomly select which event has occurred
3. **Update** - based on 2. move the model time forward to the event time and update the state of the system
4. **Repeat** - Iterate through steps 2. and 3. until some stopping criteria is met

A simple example of creating a dimer with the help of two monomers can be used to explain the Gillespie Algorithm, which is as follows:

Consider two different molecules A and B. These two monomers reversibly combine to form a dimer AB, considering the reactions are possible: A and B react to form AB, or AB dissociates to form A and B.

$$A + B \rightarrow AB$$

And the reverse reaction is given by,

$$AB \rightarrow A + B$$

Let us consider the rate of the forward reaction to be r_f and the rate of the backward reaction be r_b. And at any given time t, the number of molecules of A be N_A, that of B be N_B and that of the dimer AB be N_{AB}.

Then the rate of the total reaction, R, can be written as:

$$R = r_f N_A N_{\boldsymbol{B}} + r_b N_{AB}$$

Now that we have set up our model, we can apply the Gillespie algorithm to it.

Let us say the forward reaction has occurred and was successful, then the probability P_F of it occurring is given by:

$$P(A+B \rightarrow AB) = P_F = (r_f N_A N_{\boldsymbol{B}})/R$$

And the probability of AB dissociating into A and B, P_B, is simply 1 - P_F, given by:

$$P(AB \rightarrow A+B) = P_B = 1 - (r_f N_A N_{\boldsymbol{B}})/R$$

Now with these two probabilities known, we can either create AB by reducing the number of A and B(N_A and N_B), or create A and B by reducing the number of AB(N_{AB}).

We have now reached time t + Δt, and have performed one reaction. The algorithm will just repeat these steps for as many reactions as we would like, and simulate the stochastic results.

This example was implemented by Lewis Cole, the results of which are given below:

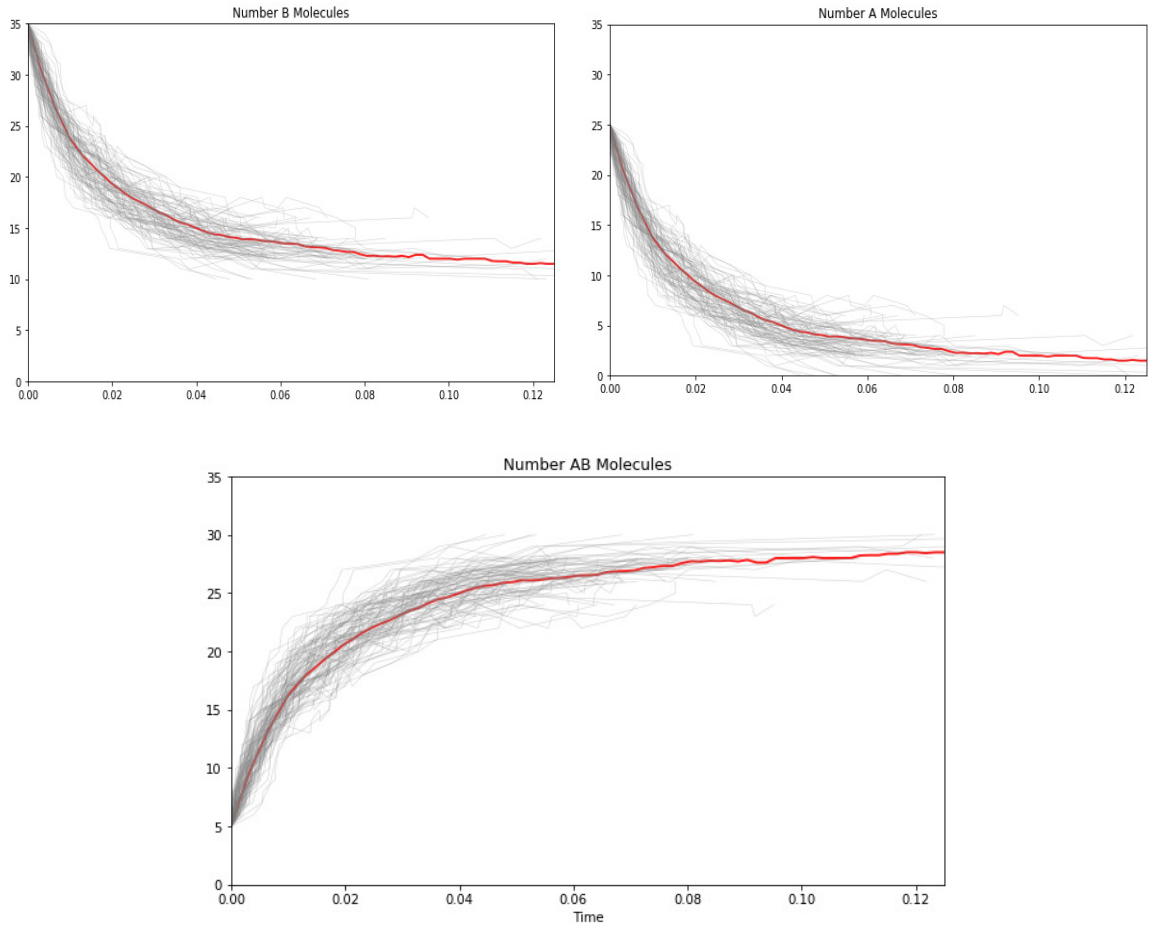


Fig 7: Various trajectories of creation of dimer and monomers along with their average.

This same principle can be applied to our various SIR models. (See *section: Implementation*).

2.3.5 Related research

In each research, it is very important to distinguish between the work that we are working on and the nature of work that has already been done. It is meaningless if one repeats or copies something that has already been accomplished, as said from the technological standpoint 'Reinventing the wheel'. This very concept acknowledges the need to make use of existing data, knowledge and research, given there already is a sea of research to choose from. Moreover, it is untrustworthy and irrelevant if people cannot relate to existing research and knowledge present in the world.

The very first distinguishing factor in this project is the approach, to use mathematical based modelling rather than agent based modelling, the second difference is the use of machine learning to predict the future of the pandemic. Thus, for my final step of this background research and literature review, I will talk about this existing area of Agent based modelling and the use of machine learning algorithms. Although there are a number of research papers and articles which use the above two methodologies, I would like to discuss the

articles, (1) ‘Measuring and Preventing COVID-19 Using the SIR Model and Machine Learning in Smart Health Care’ (Journal of healthcare engineering, 2020), and (2) ‘An agent-based approach for modelling dynamics of contagious disease spread’ (Liliana Perez and Suzana Dragicevic, 2009).

Coming to the first research article, on one hand this article is very comprehensive due to its relevance in making use of data and understanding mathematical modelling. They correctly constructed a model, “A model measuring and preventing the continued spread of COVID-19 is urgently required to provide smart health care services.”. They use the SIR model for citizens in Saudi Arabia, “The proposed system predicts whether COVID-19 will spread in the population or die out in the long run” (Journal of healthcare engineering, 2020). This shows the similarities in my approach to theirs.
On the other hand, once the model is ready, they next use machine learning principles to estimate the future, “Mathematical analysis and simulation results are presented here as a means to forecast the progress of the outbreak and its possible end for three types of scenarios: no action, lockdown and new medicines”. While I used optimization techniques to see the effects of the lockdown to help make better decisions. Both the methods seem similar yet serve different purposes. The paper predicts the future based on current decisions while my research helps making decisions thus avoiding havoc in the future.

The second research article does not guide my research in any way, it simply shows a different perspective on how we can view an epidemic. The objective of this study is “to develop an agent-based modelling approach that integrates geographic information systems (GIS) to simulate the spread of a communicable disease in an urban environment, as a result of individuals' interactions in a geospatial context.”. This study models a SEIR model, and applies Agent based modelling principles to “provide insights into the application of the model to calculate ratios of susceptible/infected in specific time frames and urban environments”. This paper studies behaviors of individuals from the bottom up. My study on the other hand does a top to bottom approach (see section *6.3.1 Methodology selected*).

Combining the knowledge and ideology from these topics in the related research article sets a parallel methodology to this project. The Journal of healthcare engineering article follows the results of the simulation and discovers, “Based on the SIR model, the pandemic will most probably be controlled by June 2021”. In parallel, this paper highlights what the research agenda of the future work on this project would be (see *section 6 Discussion and future work*)
This article has brought a lot of understanding of grasping the concepts of the SIR model and the measures we can take, and shown relevance to this project's approach. It was thus very beneficial for me to see these models and the results they show.

SOFTWARE ANALYSIS

As I was planning on how to carry on my research during the initial brainstorming of the subject, the very next step was to decide on the software(s) that I would use to create my models and simulations. From an entire ocean of software packages and tools to select from, I could not just randomly select a tool which “seems” useful. Thereby, I performed a detailed comparison between the most trusted tools for the purpose. (see *Table 2: Software selection analysis*)

Programming Language / Software	**Last Update(Year)**	**Advantages**	**Disadvantages**	**Personal Comments**
Python	2021	A very large package library for various analytical and statistical analyses.	Very slow. Dynamically typed. Need library packages installed on the computer to run.	Proficient in language
MATLAB	2021	Supports modelling approaches, statistics, ODE solvers and other mathematical tools	Interpreted language, and thus very slow. Not free.	Proficient in language
Java 8	2014	Excellent frameworks for data science/analytic s. Fast compilation time.	Syntax is not very user friendly. Not used very often for data analytics.	Used java only for software development
Javascript	2015	Fully browser based. Good data visualization library(d3).	Internet connection required. Hard to use compared to other language libraries.	Prefer python over JS in order to do data science
R	2021	Ease of use. Wide selection of frameworks.	Very slow. Steep learning curve.	Have no prior knowledge of the language

Table 2: Software selection analysis

Of the many relevant languages and tools, I choose to use Python for my project. The entire project is completed in Python and its various libraries. Originally the debate was between Python and MATLAB, mainly because I am proficient in both when it comes to solving scholastic problems. However I soon realised how widespread python is when it comes to data science. Also it made more sense to use Python for the MSc Computing degree rather than MATLAB.
Java and Javascript also made sense given the degree programme, yet my experience with Java is confined to software development and that of Javascript for Front end web. It would have been an interesting challenge to learn either of the two. But alongside my experience in Python, it only made more sense given its relevance in the data science community.
R although comes par with Python in terms of solving mathematical problems, but I have absolutely no prior knowledge about R, and given the complexity of the problem I have in hand. I felt it was too big a risk to learn a language in less than 3 months and apply working knowledge into simulating models.

On the technical side, there are a number of reasons for it. Primarily the vast sea of libraries it offers to solve mathematical and statistical problems. It is very interesting to see how there is a Python library for almost every problem out there. No other language offered these many resources.

NumPy being the most useful library out of all for modelling the epidemic, while Matplotlib for visualising the model.
NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more. Some also call it an alternative for MATLAB as both essentially use multidimensional arrays to solve mathematical problems. One can argue why we need arrays when we can use a Python list, the answer is simple as NumPy arrays are over 50x times faster than a Python list. Which is extremely beneficial when we are solving complex mathematical problems.
Matplotlib is a library extension for NumPy, it is a plotting library. It was designed to be similar to MATLAB, with the ability to use Python and make it open sourced and free to use. There are also a large number of third party packages extending the functionality and usability of the library, this also includes several high level plotting interfaces like seaborn and ggplot. Also, having learnt this package in CMT218 Data Visualisation, it was the right choice to select this.

Coming to the control side of the problem. Alongside NumPy and Matplotlib, I needed an optimisation algorithm in order to understand the effects of different lockdowns on the model. There are about 600 python repositories which focus only on optimisations. Going through every single one is not logical, yet to find the one which would suit the purpose of this project. I found the GEKKO optimisation suite to be the right tool for this purpose. As mentioned in the article “GEKKO Optimisation Suite”, Logan D.R. Beal said, “GEKKO greatly

facilitates the development and application of tools such as nonlinear model predictive control (NMPC), real-time optimization (RTO), moving horizon estimation (MHE), and dynamic simulation. GEKKO is an object-oriented Python library that offers model construction, analysis tools, and visualization of simulation and optimization".

And finally moving to the Monte Carlo simulations, the Gillespie algorithm (see *section 6.4 Monte Carlo Simulations*) is written using OOP concepts of Python, NumPy and Matplotlib.

I also made the decision to write all my code using a code editor (VS Code) rather than the Jupyter notebook, as now my code can be run on any device that has Python installed. Thereby removing the need to have Anaconda installed. Overall, with its vast library selection and ease of use Python proved to be the perfect language for all my models, simulations and optimisations.

REQUIREMENTS

To meet certain expectations and have the results of the model functioning properly, we must first make sure that a number of criteria are set out to be implemented. And these criteria need to be met before running the simulations on any device. They can well be divided into Functional and Non-Functional requirements as given below:

Functional requirements

Anyone who wishes to view the simulations must have these installed on the PC.

1. Python 3.8.1
2. NumPy 1.18.3
3. Matplotlib 3.4.2
4. Regular expressions 2.2.1
5. SciPy 1.6.3

With these softwares and packages installed on the computer, you can run the simulation on any code editor.

Non-Functional requirements

1. Code Readability - The code must be clean without junk data and must be commented clearly wherever applicable.
2. Performance - The application should collect the data (if any), process the code fast and efficiently using minimal resources and time.

IMPLEMENTATION

I had already chosen Python to be my programming language for the project. In order for me to successfully create my models and see simulated results, I needed to first brainstorm on how I was going to go about making the models. I began by organising my research and dividing the programming section into three different parts thereby narrowing areas based on my background knowledge and assumptions.

The file structure is also divided subsequently into three folders, Epidemic modelling, Control theory and Monte Carlo simulations.

5.1 Epidemic modelling

This folder contains all the source codes for the epidemic models. It contains the models for SIR, SEIR and SEIRS model.

Given that I had to make 3 models, all of which would require me to solve ordinary differential equations. It made more sense to make a different class for solving the differential equations rather than rewriting the code over and over again. This method uses Inheritance which was taught in CMT219 Algorithms, Data structures and Programming. The file ODESolver.py contains code for a differential equations solver which is inherited by other files which contain the code for the modelling. In this section I will explain the SIR model, as the implementation for the rest of the models are exactly the same with the difference of only the differential equations.

Ordinary Differential Equation Solver

This contains two classes, one parent class 'ODESolver' and one subclass 'ForwardEuler' which defines the method with which we will solve the differential equations.
This file can compute any set of ordinary differential equations and thus is very general.

The parent class, or the superclass, when called from the SIR.py file. Takes in model data and equations, solves it and returns the solutions. As shown in the workflow below.

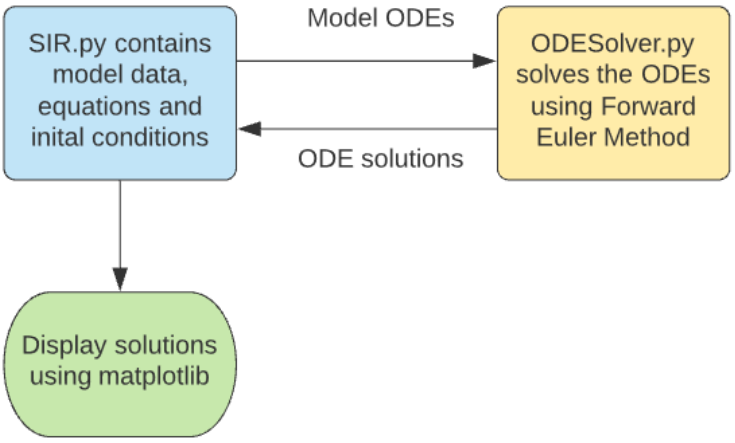


Fig 8: Basic workflow.

The solver contains four functions, the first one is the initialization constructor, the second one the 'advance' function, or any classes inheriting from the parent class must implement the advance() function, a function to take in the initial conditions and stored within the class to solve the ODEs.
And finally a solve() function, one of the arguments for this function is time_points. This means we are going to solve every differential equation for a bunch of time points and this also defines the precision that we will be using. The very first time points are going to be the initial conditions set in SIR.py, and then the equations are integrated as we have a derivative on the left hand side of the equation.
For example in the equation, $\frac{dS}{dT} = -\frac{\beta SI}{N}$
Going by the definition of integration, for every small time interval we need to find the area under the graph. Calling the advance() function (Fig:11).
Then we finally return the solutions.

We then move to the subclass, which inherits from the parent class ODESolver, using the Forward Euler method; this is one of the simplest solvers for differential equations. And given the complexity of this project is not too large compared to other high end projects which require lots of data processing power. Thus, this method works perfectly given the scope of this project.
This method is based on the Taylor series expansion, and explained below:
Consider t_n to be the time at the nth time-step and the solution s_n at this time-step be equal to, $s_n = s(t=t_n)$. There is a step size which is assumed constant for simplicity in calculations and is given by $t_n - t_{n-1}$.
Then the forward euler method computes y_{n+1} to be:

$$y_{n+1} = y_n + h * f(y_n, t_n)$$

```
def advance(self):
        u, f, i, t = self.u, self.f, self.i, self.t
        dt = t[i+1] - t[i]
        return u[i, :] + dt*f(u[i, :], t[i])
```

Fig 9: advance() function following the Forward Euler method

SIR model

This also contains a class which defines our problem. This class contains two functions, first being the __init__() method or the constructor with all the initial conditions and the parameters for the differential equations.
In order to make the function more robust, we make the parameters vary with time. We do this with the help of the lambda function, if the parameter is an integer or a float, we convert it into a function as shown in fig: 12 below. In simpler terms, for every 't' we input, the output will always be 'nu'. This is done for all the parameters for the differential equations.

```
self.nu = lambda t: nu
```

Fig 10: Converting parameter into a function

The second function is the __call__() method, this is used to make the class object callable. The main function of this method is to evaluate the function at the current time step. This function returns a numerical array in Python which essentially contains all the differential equations.
We then finally have the driver code where we run and plot the results.
This is the exact same for the SIER and SIERS files, with the only difference being the initial conditions, the parameters and the ordinary differential equations.

5.2 Optimal control theory

To optimally control the models, we have in place the lockdown measures. These measures can be controlled and we can see the different results for each of the policies we put in.
In order to do this, I have used the GEKKO optimisation suite, as mentioned in the previous sections.

After setting up the initial conditions and the parameter values, we then use the GEKKO function to enter the differential equations in an array.
I also inserted some very small time steps in the functions.
In order to initialize the simulations, we then use IMODE=7, this is the sequential simulation as explained in the section above.
And we finally plot these.

```
m = GEKKO()
u = m.MV(0,lb=0.0,ub=0.8)

s,e,i,r = m.Array(m.Var,4)
s.value = s_initial
e.value = e_initial
i.value = i_initial
r.value = r_initial
m.Equations([s.dt()==-(1-u)*beta * s * i,\
             e.dt()== (1-u)*beta * s * i - sigma * e,\
             i.dt()==sigma * e - nu * i,\
             r.dt()==nu*i])
```

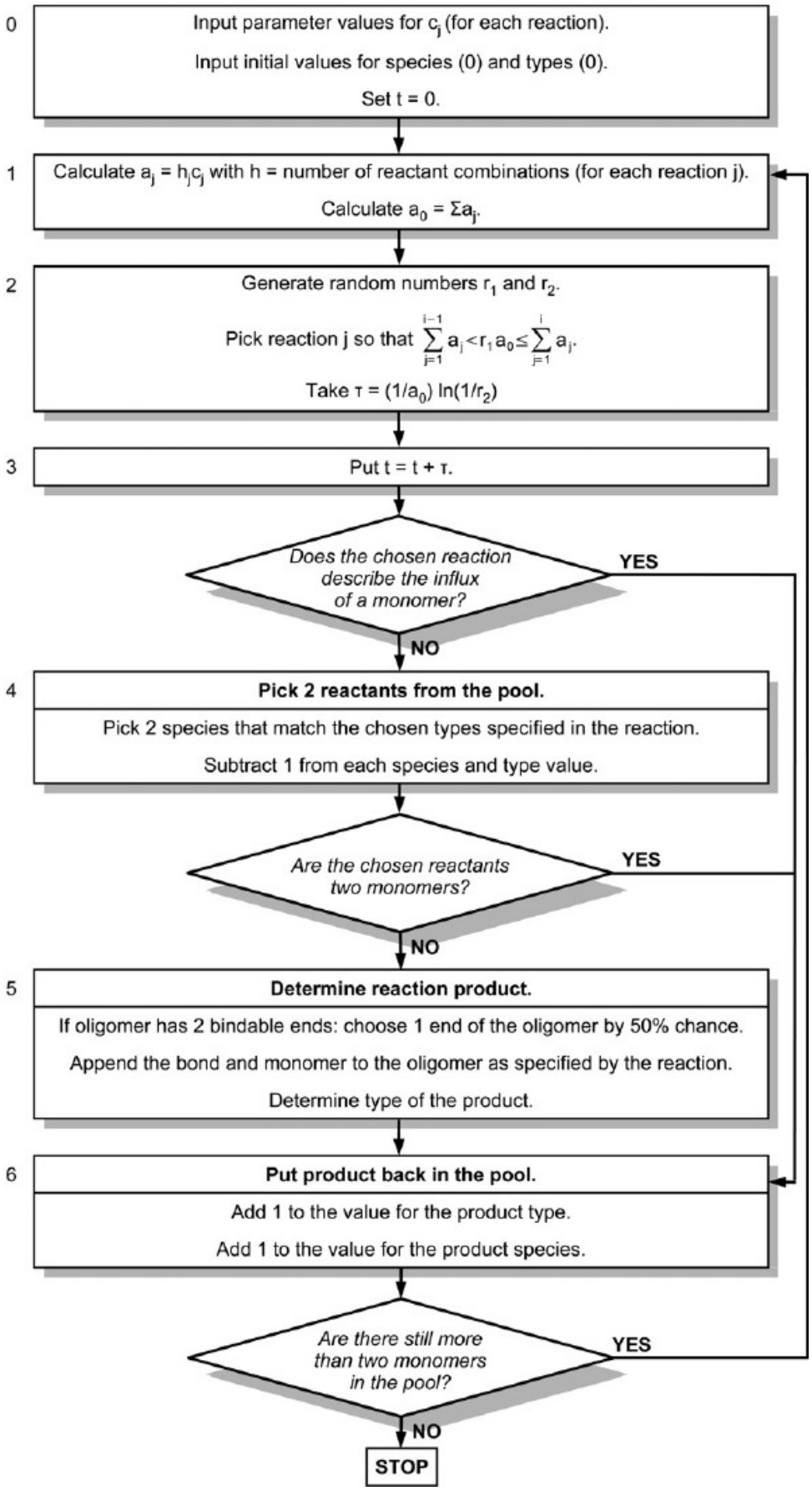
Fig 11: Using the GEKKO package

After this we optimise this to adjust the lockdown measures to maintain within the healthcare policies.
The optimization is done using the IMODE=6, this is done using the central limit theorem(CLT).
Akhilesh Ganti defines CLT to be, “In probability theory, the central limit theorem (CLT) states that the distribution of a sample variable approximates a normal distribution (i.e., a “bell curve”) as the sample size becomes larger, assuming that all samples are identical in size, and regardless of the population's actual distribution shape. ”
Thus using this probability theory technique we are able to simulate the results and get the best solution with various lockdown policies in hand. As displayed in the results section below.

5.3 Monte Carlo simulations

This as mentioned in the previous sections is done with the help of Gillespie algorithm.

The above flowchart is made by Prof. John Ralph at the University of Wisconsin. It is for the monomers example discussed previously. This has helped me a lot in a better understanding on how we can solve the problem.



#

Fig 12: Flowchart for Gillespie algorithm

This section is also solved with the help of classes. We have 2 classes, one for the SSA and the other for the SSA model.

Class SSA and Class SSAModel

SSA stands for stochastic simulation algorithm.
The first class is responsible for solving the algorithm, it follows these steps:

1. Generate pseudo random numbers
2. Indefinite generator of direct method trajectories, this is a trajectory based solution given the Gilespie algorithm generates a solution trajectory for the problem.
3. Indefinite generator for first reaction trajectories, this will evaluate the reaction time, the next reaction time and the reaction itself.

The second class is responsible for initializing the model, following the given steps:

1. Initialize the model.
2. Check breakout of trajectories, i.e; see if there are any more reactions to do, if not breakout from the loop.
3. Validate and invalidate model reactions, to see all the possible and the impossible reactions.
4. Clear the trajectories, finally reset the species or parameters to the initial conditions. Or to go back the steps as shown in the flowchart for the Monomers problem.(Fig: 14)

Driver code

This is the part where we initiate the SSA class and the SSAModel class with initial conditions and then finally plot the results.
We give the initial conditions, and the sojourn times(or mean waiting time). The propensity functions, which describe the probability while the reaction rates describe the changing rate. And finally the change in the parameters or species for every propensity.
These are given to the SSAModel which will generate the model, and the results of these models are then fed to the SSA class which contains the solution to the problem.

Once we have these, we can solve them and output the results.
This section used the odeint method from scipy integrate to compute the results in order to decrease the lines of code, ease of understanding and for lower time computations.

RESULTS AND EVALUATION

Epidemic modelling

These below show the epidemic modelling results, the SIR, SEIR and SEIRS.

The three graphs are pretty common to one another, the most common things to observe are the trends.

The susceptibles are the highest when it comes to the beginning of the graph, but they slowly go down as infected start to increase showing movement from one compartment to another. And overtime, the recovered start to increase and overtake the infected and the susceptibles.

The same is the case for SEIR and the SIR models, both of these have the same trends with the only difference is the addition of the exposed compartment.

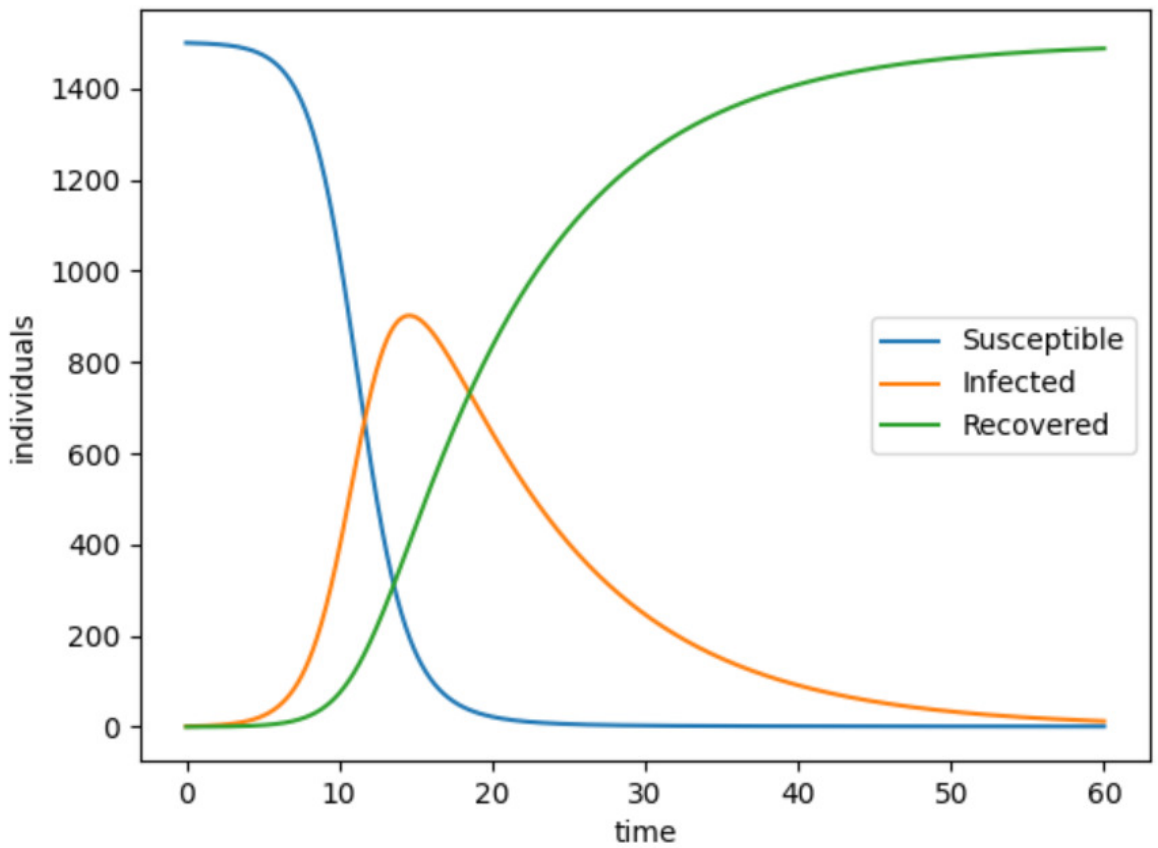


Fig 13: SIR Model

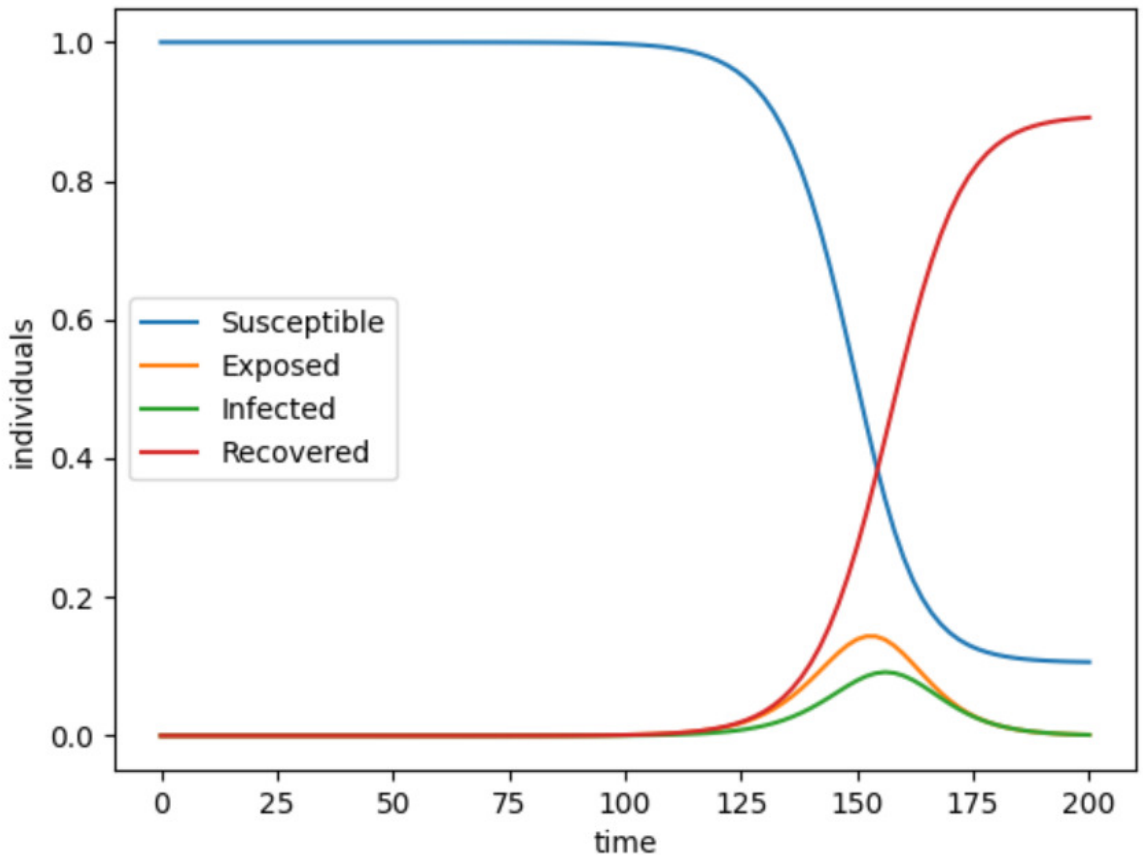


Fig 14: SEIR model

The output for the SEIRS model is a bit different as we now have a connection from the recovered back to the susceptibles. With this we can see over a long period of time, the susceptibles seem to be on the rise again. And the recovered going down, this is a much more realistic model of the current pandemic as we can see ups and downs in different compartments over time due to reasons like a new variant, vaccination policies or even lockdown measures.

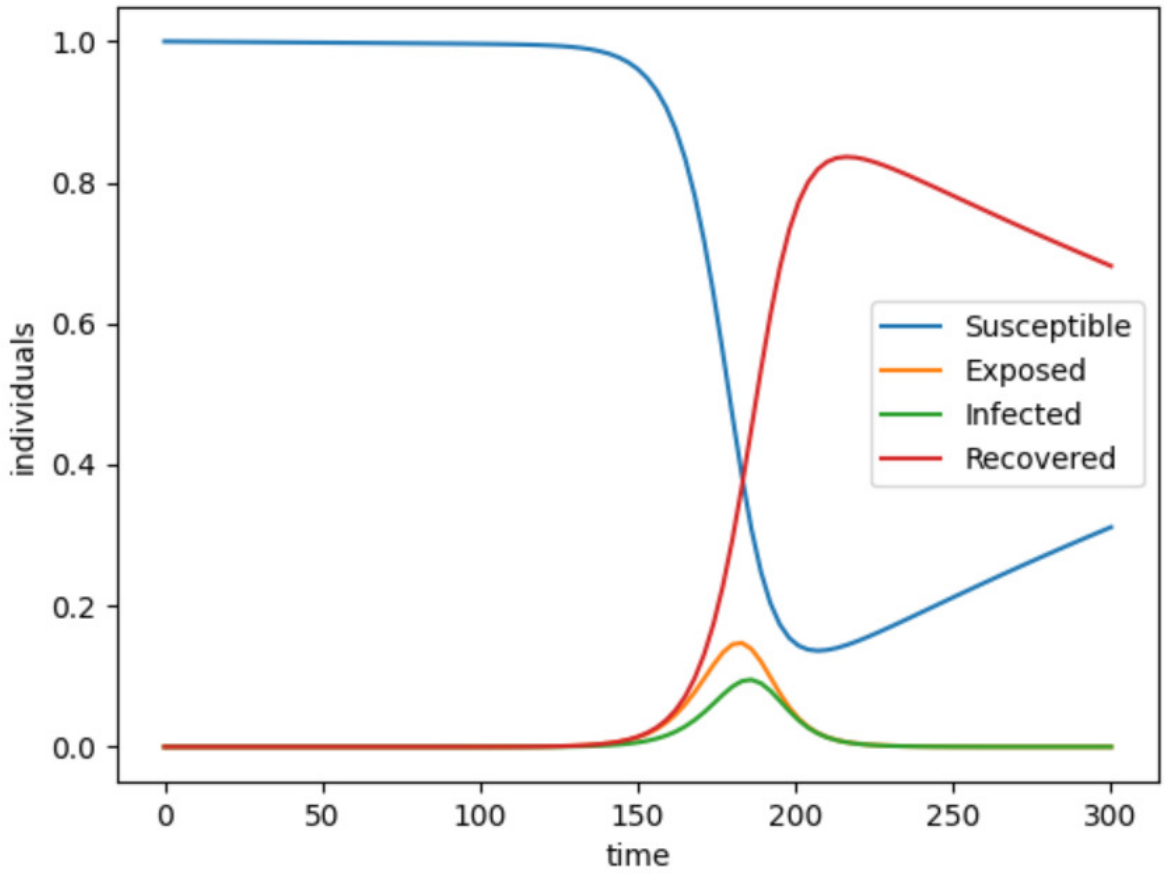


Fig 15: SEIRS model

Optimal control theory

Below are the results of the different lockdown measures that can be implemented to see the various outcomes.

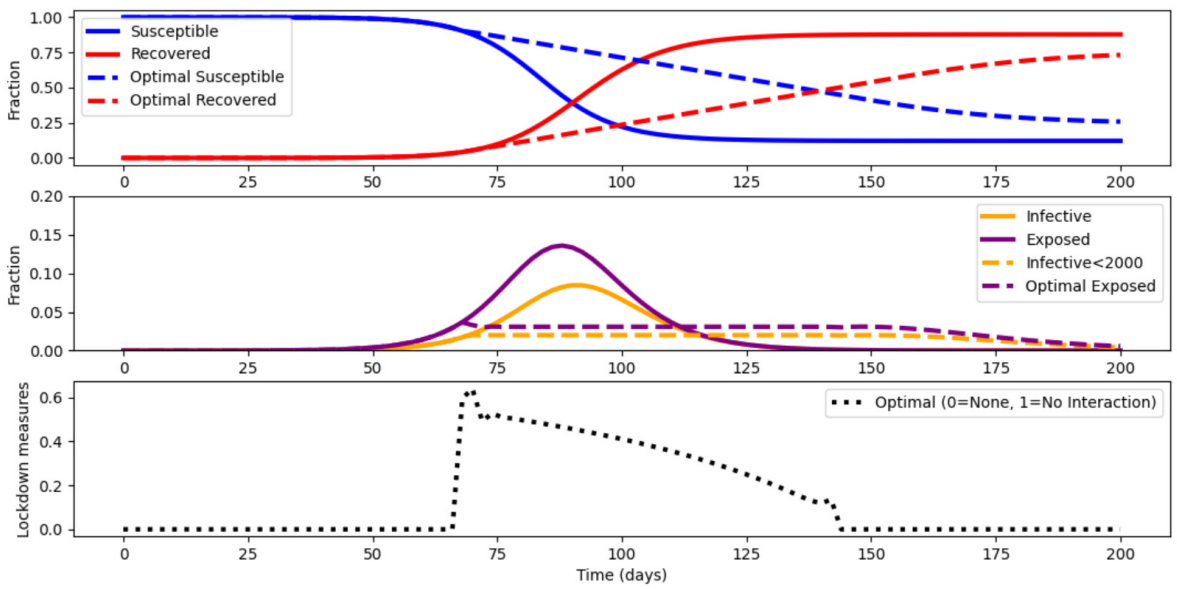


Fig 16: Strict lockdown measures

With Fig 18: Strict lockdown measures, we can see the as the pandemic starts a lockdown has been implemented, that too a very strict one, with such a lockdown being in place, we can see the optimal exposed and infected seem to be a constant value and eventually the pandemic dies out. The length of the pandemic might be more than without the lockdown but we save more lives and less people can get infected.

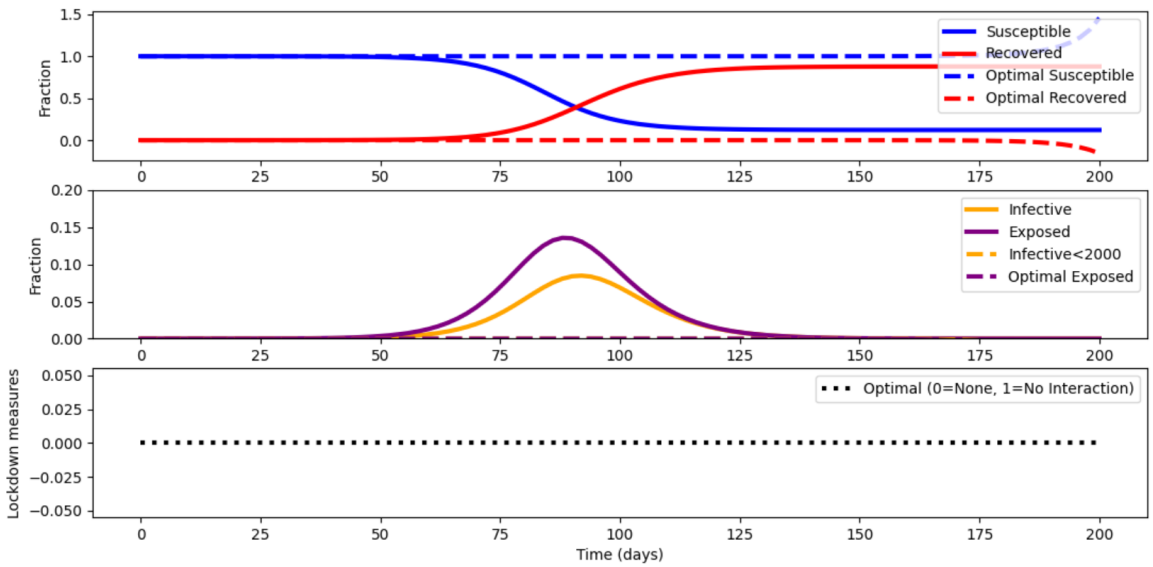


Fig 17: No lockdown measures

On the other hand with no lockdown measures in hand as shown above, Fig 19. We can see it has no impact on the optimal solution curves, they simply remain negligible. The pandemic dies out due to herd immunity with a lot of casualties and with a lot more people getting infected due to this.

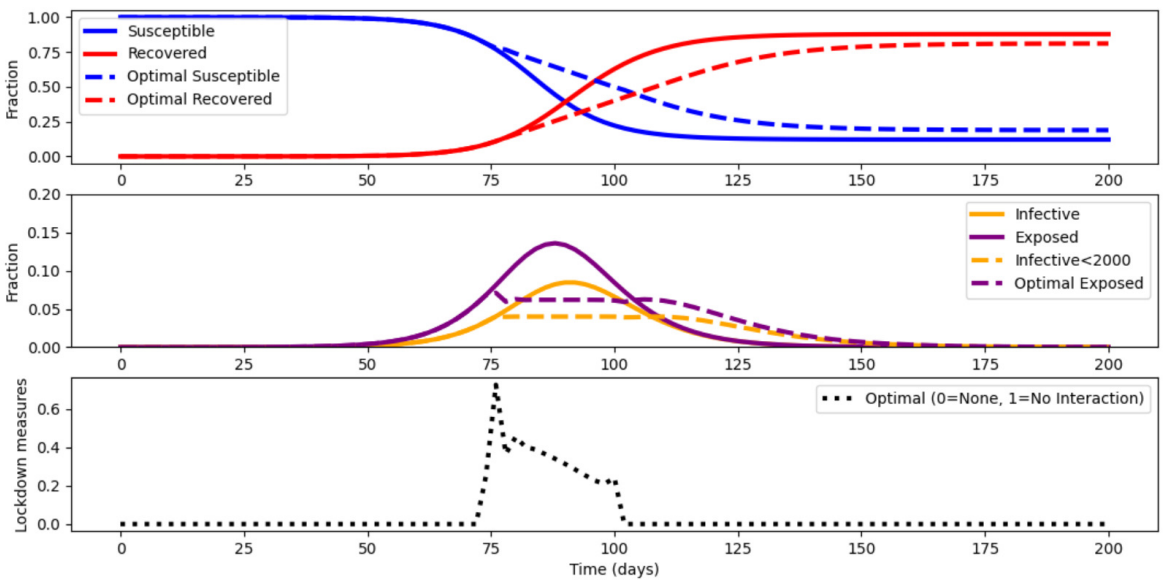


Fig 18: Moderate lockdown measures

Finally with moderate lockdown measures, we can see the lockdown is imposed when the pandemic starts and when the situation seems under control. The lockdown measures are taken away, this can be seen with a small increase in the optimal curves. And when it looks like things can get out of hands soon, the lockdown measures are then again put into place. This type of lockdown measures are usually seen on a realistic scale rather than the first or the second situations.

Evaluating these results, we can successfully predict what the pandemic can prove to be if we implement a certain lockdown measure, there can be further changes to the parameters of the code in order to see more different results in the lockdown measures.

Monte Carlo simulations

Below shows the results of the monte carlo simulations, this when compared to Fig 15, shows similar or same trends proving both the methods gave the same results. This in turn checks for the correctness of all the models that we have created are correct and results seem trustworthy.
The orange part in the graph below shows the various reactions that have been simulated and the black lines show the mean of all the reactions.

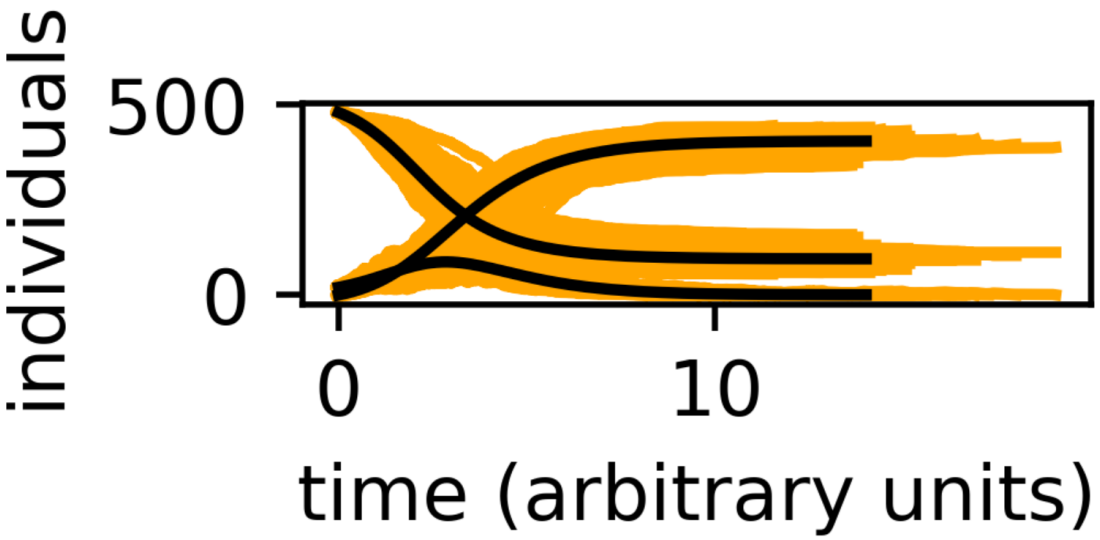


Fig 19: Monte carlo simulations

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Stopping an emerging disease like the COVID-19 in its earliest stages is vital to stop a pandemic. Mathematical modelling as done in this research, or similar provide a solution to quantify or minimise risks and assess the effectiveness of public health policies before the next event occurs.

To conclude the project, it was an overall success with a few incompletions as justified in the evaluation and elaborated in future work below. Rephrasing the problem statement as identified in the Introduction, the aim was to "Build a successful SIR(and its variants) model, and simulation to help the healthcare workers in making better decisions". Summarising from results and the evaluations, I have built three models, the SIR, SEIR and the SEIRS, along with simulations and optimised them to find impacts of a lockdown on the system. And finally checked with its robustness using the Monte Carlo simulations.

7.2 Future work

Now that the project has come to an end, as concluded it was a success. Although it has a few limitations, or additions that could make the project a complete well rounded piece of research. Now with any project, there can be a number of things that can be done better or we can make any application more convenient and add in more features.

There are two important things that I would like to focus on when it comes to the context of smart healthcare systems and how computer science can help in the medicine industry: (These are expected to be implemented considering I had 3-6 more months for the project)

First, as discussed in the *Section : Methodology selected*, we can see there can be a number of other control policies that could be implemented with the models. Alongside lockdown policies, below are a few other policies that could be optimised in the project:

1. Vaccination policies. For eg: In the UK, the rate of vaccine delivery was dependent on the rate of supply of the vaccine rather than the workforce capacity(Dept. of Health and Social care). The department also said, "An extended interval between vaccine doses together with initial prioritisation of the first vaccine dose would increase the flow of vaccine supply in the short term. This will allow for more first doses to be delivered to more people earlier.".
2. Effects of new Medicines. This is done by adding in new parameters to the models which mainly reduce the number of deaths as new medicines tend to be more efficient in healing patients.
3. Effects of a new Variant on the Model. This turns out to be a bit more complicated as now we have a tree in the SIR model with two infected (I_1 and I_2), and two recovered (R_1 and R_2) compartments.

And secondly, the use of AI and Machine learning algorithms. Time series forecasting algorithms using the Scikit-learn library in Python could be used to reduce the negative impacts of the pandemic and accelerate the recovery phase. These algorithms could be used to see how the model would behave in the future with different trends.

Given I am a MSc Computing student, this might seem too far fetched for me. But with the additional time and right resources, I believe the above two are achievable. This would allow for a better and more realistic model to operate and thus give much more accurate results to help curb the pandemic.

REFLECTION ON LEARNING

This being the final stage of the MSc Dissertation, and of my masters degree, I believe reflecting upon is critically important in any work I do. This is true in any form of work from studies, work, sports or any other activity. To bear in mind the concept of 'What have I learned?', or 'What can I take away from this?'.

Firstly, I think the most important and biggest learning curves with the entirety of this research were motivation, research and self-discipline. These three factors link with one another and are needed to complete a successful project. From a reflection point of view, there were stages in the project when it was hard for me to uphold my motivation and discipline. This happened mainly at the start of the project given how the topic was completely new to me and thus took me longer to get into a clear mindset, paralysing me to identify the right plan of action. But with the progression of my research, I got more comfortable with Modelling, control theory and other stochastic simulation strategies. The struggle for motivation and discipline also seemed hard to get with the ongoing COVID-19 pandemic, staying away from home, this was particularly hard given the uncertainty and taking my focus away from other important priorities. And working in such a routine, I have learnt that with such situations: help is always out there and it is very important to be consistent with a solid routine surrounding studies or work. Next time I work on a project, I would make sure that I have a more prominent work space, make myself a time plan for the week and allow for relaxing on the weekends. In all honesty, there also has been some positive effect of the pandemic, it allowed me to see the reason why researches like these are critical.

Another area that I believe connects to my first reflection is the fact that modelling and controlling is quite new to me. Being an electrical engineer, I have some experience in modelling electrical systems but none when it comes to compartmental modelling. So I had some knowledge on what the principle of modelling is based on but none about the problem I had to solve. To bridge this gap, I was extremely lucky to have support; in my case, to have a technical advisor, Dr Frank C Langbein, to give me technical feedback and guidance. To share an example, I was struggling to understand one of the models initially and I brought it up in one of the meetings and all he asked me to do was change a parameter, and that was it. He later explained how it fundamentally affected the entire model.
Furthermore, with the ongoing pandemic, the only way to communicate was via online video conferencing services. And thus came my adaptation to work and share my research online and receive subsequent feedback. This proves how well I have handled it, ensuring there were weekly meetings with my technical advisor, these were held on zoom. The results were shared via screen share as it served to be the most efficient of tools for the purpose.

Coming to a more technical aspect, I have had technical knowledge with Python, and knowledge about modelling with the help of MATLAB. But no prior knowledge of modelling with the help of Python. I would not say this was too hard to do considering my technical background, but it was certainly something new and exciting to work with. There were frameworks and principles which I did not understand at first, but after a few hours with them. And with a bit of practice, I was pretty confident.

Finally, coming to a more research based learning approach, documenting and writing are two key skill sets that I picked up with my project. Being an international student, and having not done a lot of research work before this. I could see myself slacking in these two areas which are critical to any paper or research. So I started to read research papers and concepts, mostly related to my topic, to understand how research is presented. Only in two weeks did I realise my mistake, I did not make myself small goals but just dived deep into papers, without any prior knowledge, and felt completely overwhelmed and lost as I could not connect the dots. I then started again, with a roadmap in hand with things to be done by the end of the month. I understood the problem in hand much more clearly, and thus I could plan everything else with ease. By the end of the report, I was pretty confident in presenting my work in a professional way, with a crystal clear understanding of what goes where on a paper.

Every single situation has its ups and downs, but at the end of the day it is all about how we approach it. This project has taught me that it is always better to plan things beforehand, it is okay if there are some amendments to the plan here and there. But what matters is the preparedness and the discipline towards the task. Overall, I can say I completed my MSc dissertation with success, and like I did in the past will always learn from my mistakes in order to grow and achieve heights.

REFERENCES

Katriona Shea, Ottar N. Bjornstad. 2020. The SEIRS model for infectious disease dynamics. Available at : https://www.nature.com/articles/s41592-020-0856-2

Jose M. Caricione, Juan E. Santos. 2020. A simulation of a COVID-19 based on a deterministic SEIR model. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7270399/

Mark Ditsworth. 2019. Infection Modelling - Part 1. Available at: https://towardsdatascience.com/infection-modeling-part-1-87e74645568a

GEKKO. 2021. GEKKO documentation. Available at: https://gekko.readthedocs.io/en/latest/index.html

Wikipedia. 2021. Compartmental models in epidemiology. Available at: https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology#The_SEIR_model

Nicole Gatto. 2021. Optimal control of the SIR model in the presence of transmission and treatment uncertainty. Available at: https://www.sciencedirect.com/science/article/pii/S002555642100002X

Jeremy R Huddy, Sharon Marie Weldon, Tim Painter. 2016. Sequential simulation (SqS) of clinical pathways: A tool for public and patient engagement in point-of-care diagnosis. Available at: https://www.researchgate.net/publication/308093975_Sequential_simulation_SqS_of_clinical_pathways_A_tool_for_public_and_patient_engagement_in_point-of-care_diagnostics

J. Jamie Hernandez, Mohan Srivastava. 2021. One Step at a Time: The Origins of Sequential Simulation and Beyond. Available at: https://link.springer.com/article/10.1007/s11004-021-09926-0

Simon Spencer. 2007. Stochastic Epidemic Models for Emerging Diseases. Available at: http://eprints.nottingham.ac.uk/11132/1/FinalThesis.pdf

Saad Awadh Alanazi, Madallah Alruwaili. 2020. Measuring and Preventing COVID-19 Using the SIR Model and Machine Learning in Smart Health Care. Available at: https://www.hindawi.com/journals/jhe/2020/8857346/

Will Wenton. 2021. Monte Carlo simulations. Available at: https://www.investopedia.com/terms/m/montecarlosimulation.asp

Mathieu Génois, Christian L. Vestergaard. 2015. Temporal Gillespie Algorithm: Fast Simulation of Contagion Processes on Time-Varying Networks. Available at : https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004579

IBM. 2020. Monte Carlo simulations. Available at: https://www.ibm.com/cloud/learn/monte-carlo-simulation

Palisade. Monte Carlo Simulations. Available at: https://www.palisade.com/risk/monte_carlo_simulation.asp

Jeff Gore. 2014. Stochastic Modelling. Available at: https://www.youtube.com/watch?v=EXBO08-78IU

Ivan. T. Dimov. 1992. Monte Carlo algorithms: performance analysis for some computer architectures. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjx6Ljr3IP0AhWGQ0EAHW4jAOIQFnoECAsQAQ&url=https%3A%2F%2Fcore.ac.uk%2Fdownload%2Fpdf%2F82000275.pdf&usg=AOvVaw1IU1HBCV3jQx70E3hYSlM5

Reka Howard. 2020. The local stability of a modified multi-strain SIR model for emerging viral strains. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0243408

Lewis Cole. 2020. Gillespie Algorithm. Available at: https://lewiscoleblog.com/gillespie-algorithm

A.S.M. 1945. Medicine and mankind. Available at: https://www.nature.com/articles/156129b0

Dept. of Health and Social Care, UK. 2021. Optimising the COVID-19 vaccination programme for maximum short-term impact. Available at: https://www.gov.uk/government/publications/prioritising-the-first-covid-19-vaccine-dose-jcvi-statement/optimising-the-covid-19-vaccination-programme-for-maximum-short-term-impact

MIT. 1998. Forward and Backward Euler Methods. Available at: https://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations_Notes/node3.html

Akhilesh Ganti. 2021. Central Limit Algorithm. Available at: https://www.investopedia.com/terms/c/central_limit_theorem.asp

Juliana Tolles, ThaiBinh Luong. 2020. Modelling epidemics with compartmental models. Available at: https://jamanetwork.com/journals/jama/fullarticle/2766672

Lawrence Evans. An Introduction to Mathematical Optimal Control Theory. Available at: https://math.berkeley.edu/~evans/control.course.pdf

Emanuel Todorov. 2006. Optimal Control Theory. Available at: https://math.berkeley.edu/~evans/control.course.pdf