Report on the paper "Time scale theory on stability of explicit and implicit discrete epidemic models: applications to Swine flu outbreak" by Yeni et al. (2024)

Matthew Gallagher - 21318676

April 2024

1 Abstract

Time scale theory has only recently been applied to the study of infectious diseases. This paper presents two discrete epidemic models, explicit and implicit, motivated by a time scale modelling approach. The majority of the paper is spent introducing these models and deriving their properties fixed points, stability and basic reproduction number. The paper then briefly applies the models to swine flu outbreak data, finding that the implicit model was the most effective at describing the data regardless of data frequency; this was the aspect I focused mostly on in this report. Firstly, I provided background context and briefly introduced the two models and their properties. After this, I coded the models into Python, along with a continuous model for comparison purposes. I then obtained the original swine flu dataset [6] that was used within the paper and prepared it for analysis. Following this, I found the ideal parameters for each model via a statistical approach (optimal AIC values). This allowed for replication of the results within this paper. I then applied the same process to an independent dataset namely, the first 60 days of COVID-19 infection in Ireland [3], also finding that the implicit model best described the data regardless of data frequency.

2 Introduction

The inspiration for this paper comes from two main sources, namely the 1927 paper "A contribution to the mathematical theory of epidemics" by Kermack and McKendrick [4], and the 2020 paper "On the exact solutions to epidemic dynamic models" by Akin and Yeni [2]. The Kermack and McKendrick (1927) paper [4] is where the continuous SIR(C-SIR) model proposed at the start of the chosen paper is obtained from. While relatively basic, especially compared to more modern disease models, the C-SIR model was remarkably accurate for its time and has provided the basic framework for almost all later epidemic models ever since. The main purpose for this model's inclusion in the paper is as a sort of benchmark to test the accuracy of the implicit and explicit models assessed in the study.

The other two models, ED-SIR and ID-SIR, are in turn heavily inspired by the Akin and Yeni (2020) paper [2]; this paper introduces time scales as a tool to convert a continuous time model to a discrete time model. To do this, they make use of two operators $\sigma(t)$, the jump forward operator and $\rho(t)$ the jump backwards operator. This tool can change a model from continuously changing in time to only changing in "jumps", e.g. once every day. The paper then applies the time scale operators to the continuous SIR model to create a discrete version of it. Motivated by the 2020 paper, the authors of the paper I selected considered 2 variants of the discrete SIR model, one with the explicit formulation and another with the implicit formulation. In the explicit formulation, the incidence rate is computed according to the population interaction in the previous time. This gives the explicit (ED-SIR) model. In the implicit formulation, the incidence rate is computed according to the population interaction at the present time. This gives the implicit (ID-SIR) model.

$$\text{ED-SIR Model:} \quad \begin{cases} \Delta S_n = -\frac{\beta(1-\gamma)}{1+\beta I_n} S_n I_n - \gamma S_n + \alpha \\ \Delta I_n = \frac{\beta(1-\gamma)}{1+\beta I_n} S_n I_n - (\gamma + \lambda) I_n \\ \Delta R_n = \lambda I_n - \gamma R_n \end{cases}$$

$$\text{ID-SIR Model:} \quad \begin{cases} \Delta S_n = -\frac{\beta(1-\gamma)}{1+\beta I_{n+1}} S_{n+1} I_{n+1} - \gamma S_{n+1} + \alpha \\ \Delta I_n = \frac{\beta(1-\gamma)}{1+\beta I_{n+1}} S_{n+1} I_{n+1} - (\gamma + \lambda) I_{n+1} \\ \Delta R_n = \lambda I_{n+1} - \gamma R_{n+1} \end{cases}$$

Where β is the transmission rate, λ is the recovery rate, α is the recruitment rate of susceptible individuals and γ is the death rate.

The majority of the rest of the paper consisted of deriving the equilibrium points of these models, as well as the respective stability of these equilibrium points. As equilibrium points occur when $(S_{n+1}, I_{n+1}, R_{n+1}) = (S_n, I_n, R_n)$, the equilibrium points of the ED-SIR and ID-SIR models are equivalent. The paper then derived the following equilibrium points and basic reproduction number.

Disease free equilibrium, $E_0 = (S_0^*, I_0^*, R_0^*)$

$$S_0 = \frac{\alpha}{\gamma}, \quad I_0, \quad R_0 = 0$$

Endemic equilibrium, $E^+ = (S^*, I^*, R^*)$

$$S^* = \frac{\gamma + \alpha}{\beta} + \alpha, \quad I^* = -\frac{\gamma}{\beta} + (1-\gamma)\frac{\alpha}{\gamma + \lambda}, \quad R^* = \frac{\lambda}{\gamma}(-\frac{\gamma}{\beta} + (1-\gamma)\frac{\alpha}{\gamma + \lambda})$$

Basic reproduction number

$$\mathscr{R}_0 = \frac{(1-\gamma)\alpha\beta}{\gamma} + 1 - (\gamma + \lambda)$$

The paper then showed that both the disease free and endemic equilibrium points of the ED-SIR model are locally asymptotically stable. Finally, it was shown that the endemic equilibrium of the ID-SIR model is globally asymptotically stable with the use of Luapunov functions. The process by which this is done was almost identical to the 2010 paper "Global stability for an SIR epidemic model with delay and nonlinear incidence" by McCluskey [5].

All of these derivations were done in sufficient mathematical detail within the paper and I did not feel the need to recite these proofs in this report. Instead, I focused most of my attention on the analysis of these models to determine which is the most effective at predicting an actual disease outbreak. However, I thought it was important to state the results relating to the ED-SIR and ID-SIR models derived in the paper, as this allowed me to code these models into Python.

3 Methods and Results

3.1 Coding the models

The first step I took was to actually code the 3 models into Python. As I later applied these models to the spread of swine flu and COVID-19 over a relatively short period of time, we assume $\alpha \approx \gamma \approx 0$. It is possible to do this as there will be relatively low non-disease related change in the population (e.g. births, deaths, migration) over a short period. This simplifies the models making them substantially easier to code.

The C-SIR model was the simplest to code, as I slightly altered the SIR model provided in the module resources to fit the model. The next two models were more difficult as they were discrete and not continuous in nature. The logic behind my code for the ED-SIR model goes as follows.

$$\Delta S_n = -\frac{\beta}{1+\beta I_n} S_n I_n$$

$$\Delta I_n = \frac{\beta}{1+\beta I_n} S_n I_n - \lambda I_n$$

$$\Delta R_n = \lambda I_n$$

Where:

$$S_{n+1} = S_n + \Delta S_n$$

$$I_{n+1} = I_n + \Delta I_n$$

$$R_{n+1} = R_n + \Delta R_n$$

For each step, I stored the value in an empty array, e.g. $S = \text{np.zeros}(t+1)$, with the first position in this array being the initial conditions, e.g. $S[0] = S_0$. The function then iterates filling the empty array up to the last position in the array, e.g. $S[t] = S_t$.

The ID-SIR model, on the other hand, cannot be solved this simply and is substantially trickier to deal with. Using the same logic as the ED-SIR code, we get:

$$S_{n+1} = S_n - \frac{\beta}{1+\beta I_{n+1}} S_{n+1} I_{n+1}$$

$$I_{n+1} = I_n + \frac{\beta}{1+\beta I_{n+1}} S_{n+1} I_{n+1} - \lambda I_{n+1}$$

$$R_{n+1} = R_n + \lambda I_{n+1}$$

This cannot be solved as we only know the initial conditions and so the S_{n+1} and I_{n+1} terms are unknown. Therefore, we must rearrange the equations to consist of known terms only. Start with S_{n+1}:

$$S_{n+1} + \frac{\beta}{1+\beta I_{n+1}} S_{n+1} I_{n+1} = S_n$$

$$S_{n+1}\left[1 + \frac{\beta}{1+\beta I_{n+1}} I_{n+1}\right] = S_n$$

multiply both side by $1+\beta I_{n+1}$:

$$S_{n+1}\left[1+\beta I_{n+1} + \beta I_{n+1}\right] = S_n(1+\beta I_{n+1})$$

$$S_{n+1}\left[1+2\beta I_{n+1}\right] = S_n(1+\beta I_{n+1})$$

$$S_{n+1} = S_n\left[\frac{1+\beta I_{n+1}}{1+2\beta I_{n+1}}\right]$$

Sub S_{n+1} into I_{n+1} equation:

$$I_{n+1} = I_n + \frac{\beta}{1+\beta I_{n+1}}\left[\frac{1+\beta I_{n+1}}{1+2\beta I_{n+1}}\right] S_n I_{n+1} - \lambda I_{n+1}$$

$$0 = I_n + \left[\frac{\beta}{1+2\beta I_{n+1}}\right] S_n I_{n+1} - \lambda I_{n+1} - I_{n+1}$$

For convenience let $I_{n+1} = x$:

$$0 = I_n + \left[\frac{\beta}{1+2\beta x}\right] S_n x - \lambda x - x$$

Multiply by $1 + 2\beta x$:

$$I_n(1+2\beta x) + \beta S_n x - \lambda x(1+2\beta x) - x(1+2\beta x) = 0$$

$$I_n + 2\beta I_n x + \beta S_n x - \lambda x - 2\lambda\beta x^2 - x + 2\beta x^2 = 0$$

Group x^2, x and constant terms, also multiply by -1:

$$x^2(2\beta(1+\lambda)) + x(1+\lambda-\beta S_n - 2\beta I_n) - I_n = 0$$

This is a quadratic; we can therefore use the $-b$ formula:

$$x = I_{n+1} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

with: $a = 2\beta(1+\lambda)$, $b = (1+\lambda-\beta S_n - 2\beta I_n)$, c $= -I_n$

Only the "+" case in the -b formula is accounted for as the "-" case produces negative values for the I_{n+1}. This is, of course, biologically unrealistic as you cannot have a negative number of people infected.

Therefore, as we now know I_{n+1}, we can solve for S_{n+1} and R_{n+1} using the above formulae. My code for this model consequently used similar logic to the ED-SIR case, with the important caveat that I_{n+1} must be solved for first as this value is used to calculate S_{n+1} and R_{n+1}.

Plots of Models

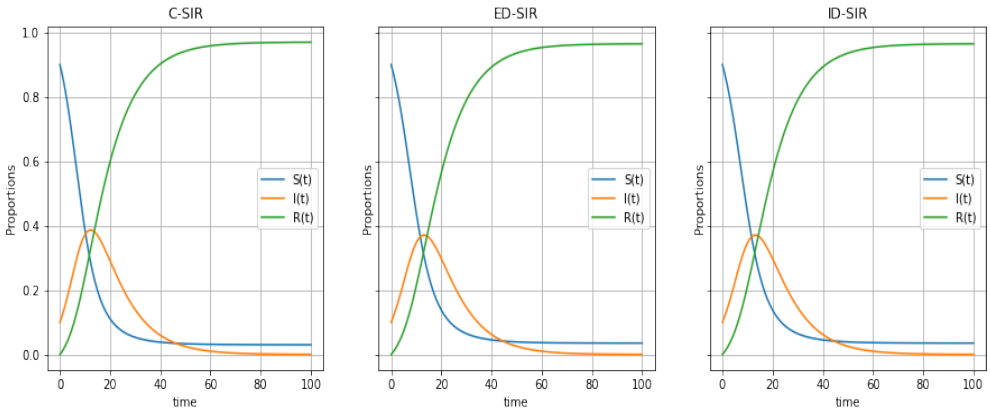

Figure 1: The 3 models with same initial conditions

The above plot shows our models with the same initial conditions.
$S_0 = 0.9$, $I_0 = 0.1$, $\beta = 0.35$, $\lambda = 0.1$.
For the discrete models $t = 100$, meaning it carries out 100 time steps whereas for the continuous model t = np.linspace$(0, 100, 10000)$, meaning the model is evaluated at 10,000 equally spaced values of t between 0 and 100.

This is an important distinction to make as the discrete graphs are made up of far less data points in comparison to the continuous graph. This is due to fact that we can always add more data points to the continuous model by simply decreasing the length of the time steps, whereas the

length of the time steps in the discrete models cannot be altered arbitrarily.

It is difficult to see any differences between the models from the above plot. Thus, I next plotted only the number of infections for each model onto the same graph to more directly compare their differences.

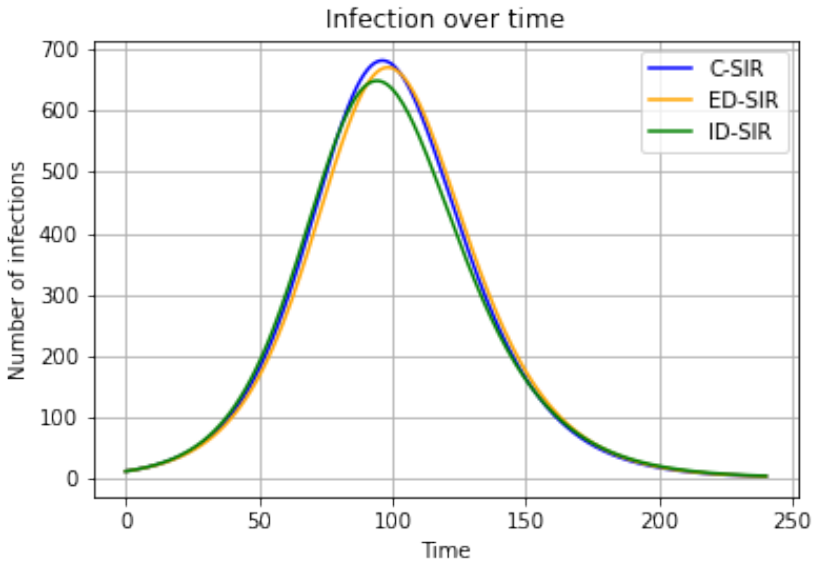

Figure 2: Infections numbers of the 3 models

Initial conditions in this graph: $I_0 = 11$, $S_0 = 18223$, $R_0 = 0$, $t = 240$.
From this graph, we see the number of infections for the ID-SIR model is shifted slightly to the left when compared to the C-SIR model and the ED-SIR model is shifted slightly to the right. This is generally what we would expect from implicit and explicit models.

3.2 Finding a Dataset

Next, I wanted to test how effective the models were at actually predicting a disease outbreak and if the paper's claim that the ID-SIR model was the most accurate was, in fact, true. The first step in this process was to gain access to the original dataset to work on. I emailed the corresponding author of the paper as they appeared to be the most logical choice for data access. They, in turn, sent me on the details of the team that carried the original study [6] on the swine flu outbreak on the Washington State University campus. After a few weeks, I was sent an Excel file with the raw data from this study.

I have attached the Excel file that I was sent with the project. The data spanned 109 days; however, similar to the paper, I decided to focus on the first 40 days of the outbreak. The data only took into account the number of new infections reported each day and not the total number of infections per day. In order to find the total number of infections there was in one day, I had to sort the data into a new array. Assuming it took 6 days to recover from the disease, I coded in Python.

```
i=0
Caught = 0
recovery = 6
Infected = np.zeros([40])
while(i<40):
```

```
    Caught += Cases[i]
    if(i>=recovery):
        Caught = Caught - Cases[i-recovery]

    Infected[i] = Caught
    i=i+1
```

For this code, we can think of an increase in (i) as an increase in one day. This code uses the number of reported cases each day and adds that number to the total number of infected people on each day, given by Infected[i]. Once 6 days has passed, the people that have recovered are removed from the Infected array. When we plot this Infected array on a bar graph, we get.

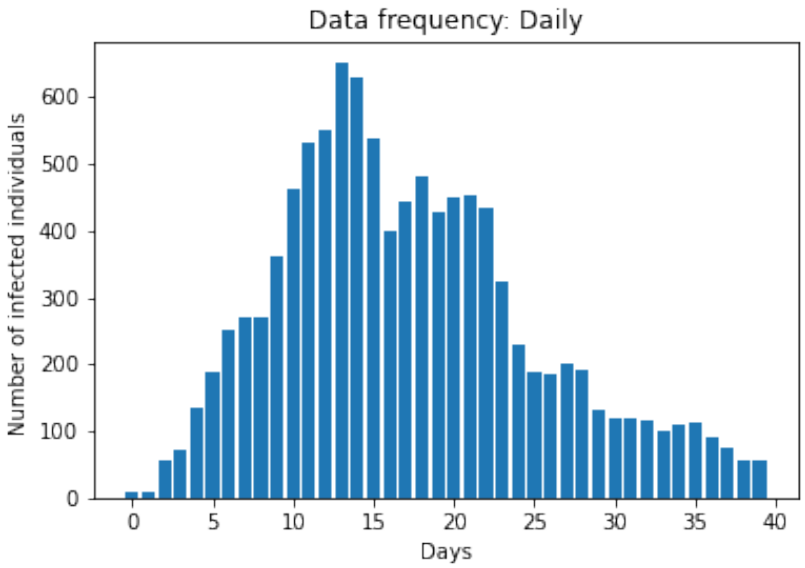

Figure 3: WSU data over 40 days

This is identical to the bar graph within the paper. To obtain the number of infections when data is collected every 2, 3 and 4 days follows the same logic, but is slightly more challenging. For example, when data is collected every 2 days, I wrote the code.

```
Infected2 = np.zeros([20])
Infected2[0] = Cases[0]
Caught = Cases[0]
i=1
j=1
while(i<39):
    Caught += Cases[i] + Cases[i+1]
    if(j==3):
        Caught = Caught - Cases[i+1-recovery]
    if(i>=recovery):
        Caught = Caught - Cases[i-recovery] - Cases[i+1-recovery]

    Infected2[j] = Caught
    i=i+2
    j=j+1
```

For this case, we set the size for the infected array to 20 rather than 40. We then add the number of reported cases on 2 consecutive days and remove the 2 days of people that have recovered to get the total number of people infected.

It is important to note in this case that we start counting our days at day 1 rather than day 0. This means that, on day 6, people who were originally infected on day 0 will have recovered; however, we cannot subtract the people that have recovered on day 5 as this would imply there is a day -1, which does not exist in the dataset. Thus, I added in an extra if statement (j==3) to account for this. The 3 and 4 days cases have an Infected array size of 14 and 10, respectively. These cases face similar issues as the every 2 days case and were dealt with in my code in a similar way. Plotting the respective Infected arrays gives the bar graphs.

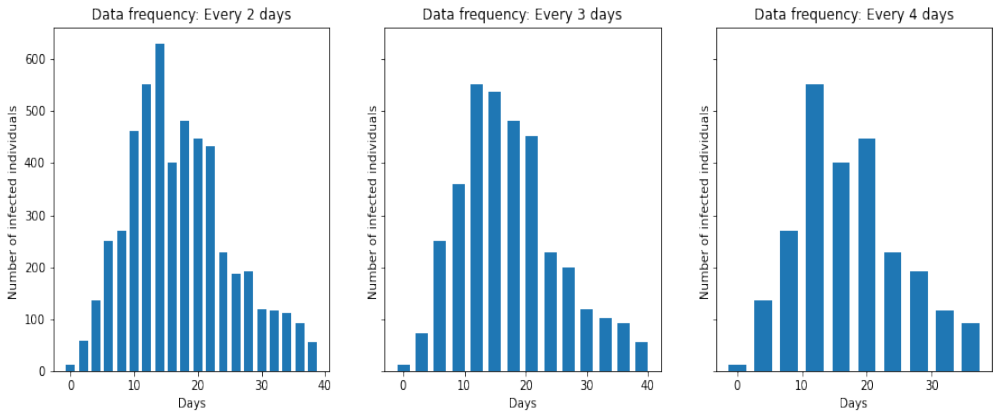

Figure 4: Data frequency every 2, 3 and 4 days

3.3 Data Analysis

The next step was to see which of the 3 models is the most accurate at predicting the dataset. To measure the "accuracy" of each model, I used the Akaike Information Criterion (AIC) [1] given by the following formula.

$$AIC = n_d log\left(\frac{SSR}{n_d}\right) + \frac{2n_d(n_p+1)}{n_d - n_p - 2}$$

Where n_d represents the number of data points. n_p represents the number of parameters estimated; for these models, it is 2 (β and λ). SSR $= \sum_{nd}(I_{model} - I_{data})^2$. The lower the AIC value, the more accurate the model fits the data.

Parameter estimation:
The initial conditions for the data is given within the Excel file $S_0 = 18223$, $I_0 = 11$, $R_0 = 0$. I then inserted these values into the models and manually varied β and λ until the models took the approximate shape of the data.

I noticed that the main effect of increasing β causes the peak of the graph to become steeper, whereas increasing λ caused the peak of the graph to become shallower. Equally increasing the magnitude of both values caused the graph to shift to the left.

I wanted a mathematical way to find which parameter values best fit the data. As changing λ and β have opposite effects, I had the idea to make a function that kept gamma constant and varied beta. I also needed to know the correct magnitude of the variables and so I multiplied both parameters by an arbitrary variable called alpha that I also varied.

```
def SearchAIC_ID(Infected, tbeta, alpha):
    i=0
    Check = 10000.0
    GAMMA = (1/6)
    while(i<61):
        BETA = 0.22
        j=0
        while(j<31):
            B = BETA*alpha
            G = GAMMA*alpha
            sol66 = ID_model(S0, I0, R0, tbeta, B, G)
            TrueAIC = AIC(sol66[1],Infected,2)

            if(Check>TrueAIC):
                Check = TrueAIC
                Betaval = B/alpha
                alphaval = alpha
            BETA = BETA + 0.001
            j=j+1
        alpha = alpha + 0.1
        i=i+1
    return ([Check, Betaval, alphaval])
```

This function searches for the ideal values of alpha and beta that best fit the dataset which is inputted into the function. The alpha argument of the function is what value of alpha we first check; we then enter the second loop and the first value for β is set. The ID-SIR model is then solved for with these parameter values and we can extract the accuracy (AIC value) of this model in comparison to the dataset. The function continues this process in the second loop for 32 *Beta* values ranging from 0.220 to 0.251. After this, we enter back into the first loop increasing the value of alpha by 0.1 and repeating the process. This is done in total 62 times; therefore, we check $62 * 32 = 1984$ different models to see which most accurately fits the data. For every new model created, we check if it is the most accurate so far (i.e. lowest AIC value). If the model is the most accurate so far, we store this AIC value as well as the associated *beta* and *alpha* values for the model. The parameters that produce the most accurate model, as well as its corresponding AIC value, are what is finally returned by the function.

We carried out this process for all 4 datasets and used an almost identical method for the ED-SIR model. To better show how these parameter values are searched for, I created animations that try to capture the process, one keeping β constant and another keeping alpha constant. Clicking the following link will bring you to these animations `https://github.com/MatthewGall3/Biology-project/blob/main/README.md`.

This method does not work for the C-SIR model, as it is too computationally expensive. However, as this model is simply used as a benchmark to compare to the implicit and explicit models, I was more comfortable with manually checking the β and λ values that best fit the data. As the C-SIR model outputs thousands of values, I extracted the values that corresponded to the same time as when the data was collected, creating a much smaller array. These new arrays were then used to calculate the AIC values for their respective models.

After retrieving these ideal β and alpha values for each case, I inserted them into the models and graphed the results.

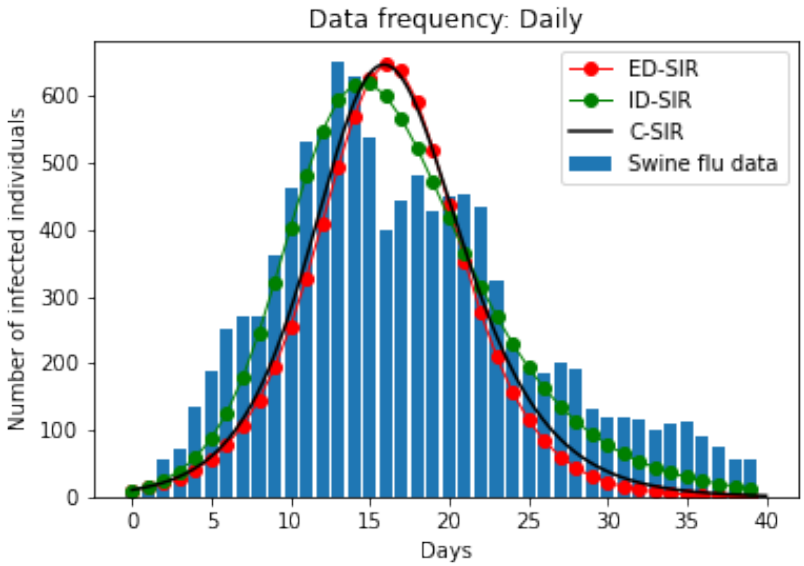

Figure 5: WSU data daily

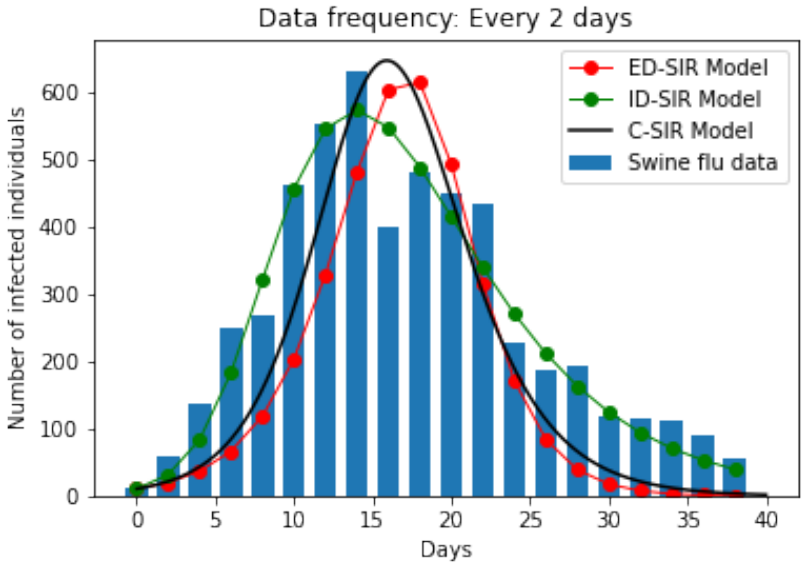

Figure 6: WSU data every 2 days

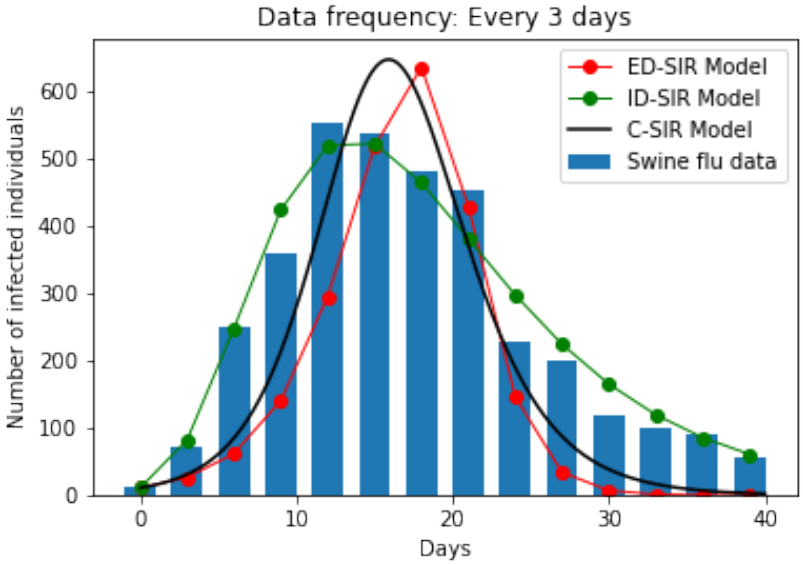

Figure 7: WSU data every 3 days

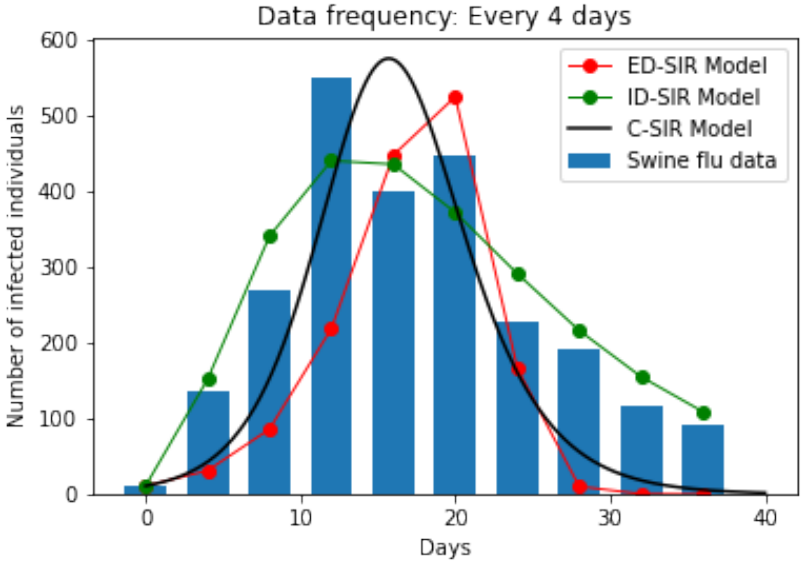

Figure 8: WSU data every 4 days

Data Frequency	C-SIR Model	ED-SIR Model	ID-SIR Model
Daily	381.450	390.507	345.224
Every 2 days	195.276	203.367	164.911
Every 3 days	135.678	144.997	108.894
Every 4 days	103.045	109.685	90.255

Table 1: AIC Values

Figures 5-8 were visually identical to comparable figures in the paper, with the optimal AIC values for each model (Table 1) also being virtually the same.

Basic reproduction number

$$\mathscr{R}_0 = \frac{(1-\gamma)\alpha\beta}{\gamma} + 1 - (\gamma + \lambda)$$

However, in this case $\gamma \approx \alpha \approx 0$. Therefore, $\frac{\alpha}{\gamma}$ (an indeterminate form) needs to be replaced by the initial susceptible population, S_0, for this data $S_0 \approx 1$. The formula for $\mathscr{R}_0$ simplifies to.

$$\mathscr{R}_0 = \beta + 1 - \lambda$$

The C-SIR basic reproduction number is calculated simply by $\frac{\beta}{\lambda}$ and stays relatively constant with $\mathscr{R}_0 \approx 1.34$. For the discrete models, $\mathscr{R}_0$ varies with data frequency.

Data frequency	ED-SIR model	ID-SIR model
Daily	1.40	1.35
Every 2 days	1.42	1.34
Every 3 days	1.47	1.32
Every 4 days	1.47	1.29

Table 2: $\mathscr{R}_0$ values

3.4 COVID-19 data

After recreating the results found within the paper, I wanted to apply the same process to a new, independent dataset. For this, I chose the first 60 days of COVID-19 within Ireland (specifically the reported positive COVID-19 tests from 5/3/2020 to 4/5/2020) [3]. COVID-19 seemed like the obvious choice as it is a highly infectious airborne disease similar to swine flu. The same assumptions were held for this data (e.g. $\alpha \approx \gamma \approx 0$), as the population of Ireland will not change significantly within 60 days. The initial parameters I chose were $S_0 = 4000000, I_0 = 4, R_0 = 0$.

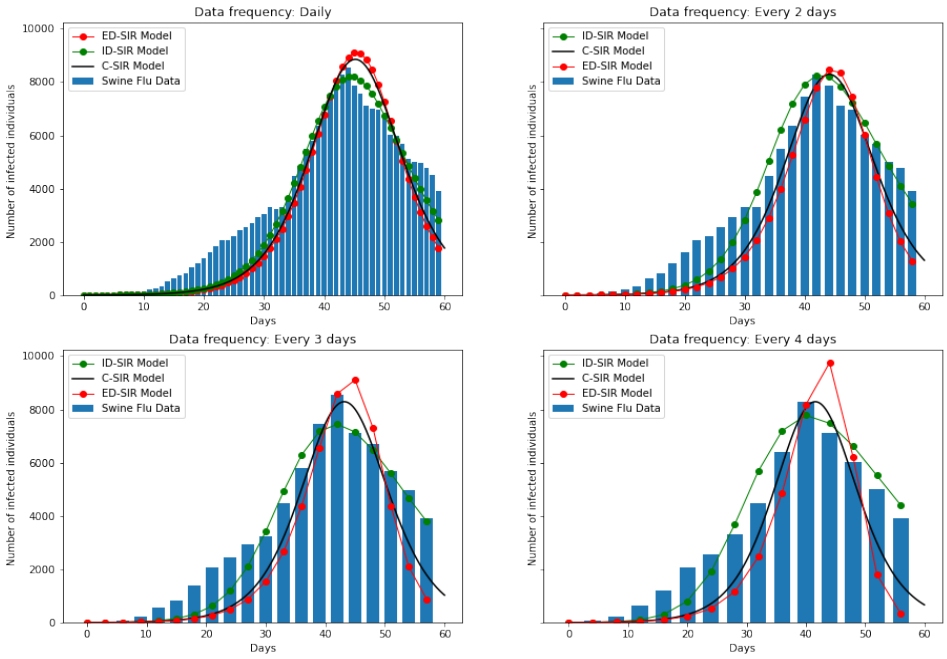

Figure 9: COVID-19 data

Data Frequency	C-SIR Model	ED-SIR Model	ID-SIR Model
Daily	829.342	840.923	798.953
Every 2 days	422.223	433.227	396.697
Every 3 days	286.446	296.311	266.017
Every 4 days	217.734	229.087	202.633

Table 3: AIC Values

Data frequency	ED-SIR model	ID-SIR model
Daily	1.22	1.19
Every 2 days	1.24	1.18
Every 3 days	1.28	1.17
Every 4 days	1.32	1.17

Table 4: $\mathscr{R}_0$ values

4 Conclusion

A key outcome from this report was the successful replication of the swine flu data results obtained in the paper, with approaches that I developed despite minimal methodological information. Specifically, visually identical graphs (Figures 5-8) and highly similar AIC values (Table 1) to the original paper were obtained. This agreed with the original findings that the order ID-SIR, C-SIR, ED-SIR of the goodness-of-fit of the models remains the same for each data subset. Having decided to apply this process to an independent dataset (the first 60 days of COVID-19 infections in Ireland)[3] not in the original paper, I also found that the ID-SIR model best fit this dataset (Table 3). This reinforces the assertion that the ID-SIR model is optimal for describing epidemiological data that is discrete in nature.

One area I expanded upon, which was only briefly touched upon within the paper, was the $\mathscr{R}_0$ values of each dataset. The paper briefly stated that $\mathscr{R}_0 \approx 1.35$ for the ID-SIR model. Using the parameters obtained through parameter estimation, I calculated the $\mathscr{R}_0$ values for the discrete models at all data frequencies (Table 2). My findings agreed that $\mathscr{R}_0 \approx 1.35$ for the ID-SIR model when data frequency was daily. However, when the frequency of data was decreased, the $\mathscr{R}_0$ values also decreased. On the other hand, the ED-SIR model had a greater $\mathscr{R}_0$ value for the daily case and, when the data frequency decreased, the $\mathscr{R}_0$ value increased. Analysis of the COVID-19 data also produced similar findings (Table 4).

These results make sense when one considers the intrinsic differences between the models. As the ED-SIR model calculates the incidence rate using previous values, its predictions will lag behind the ID-SIR model which uses present values. Thus, the ID-SIR model will need a larger $\mathscr{R}_0$ in order to accurately capture the peak of the data. This leads the ID-SIR graphs having more jagged peaks in comparison to the ED-SIR graphs which more smoothly fit the data. This effect is compounded with less data points, thus the $\mathscr{R}_0$ values diverge from each other at lower data frequency. This may explain the greater accuracy of the implicit model as it interpolates, rather than extrapolates, which is generally considered to be more accurate.

There are some limitations to the models that I produced, with perhaps the most obvious being the assumption that $\alpha \approx \gamma \approx 0$. While this assumption is reasonable over a short time frame, it may break down over a longer period of time as people will eventually be added or removed from datasets (e.g. due to births, deaths, migration). Another limitation is the assumption of lifelong immunity which may be true for some diseases but is largely untrue for the swine flu and COVID-19 datasets that these models were created to analyse. Therefore, the models should only really be used over a short time period (i.e. less than a year) to be accurate.

Accordingly, an interesting area of study would be to modify the models to also be effective in the long-term. This could, for example, involve altering the models to add an additional parameter that accounts for people that have lost immunity from the disease and therefore return to the susceptible class. We would also not be able to assume $\alpha \approx \gamma \approx 0$. However, this would make the coding of the models substantially harder, particularly if the parameters have an unknown value. If the new parameters have known values, coding of the explicit model becomes much easier. However, there are still difficulties with the implicit model, as trying to solve for I_{n+1} produces a cubic equation. This would require use of a root finding method, making the process far more computationally expensive.

In summary, I found that discrete models can be used to accurately describe a disease outbreak, with these providing a suitable alternative to continuous models. Such models may describe epidemiological data better due to the discrete nature of the data. This concept is supported by the observation that the implicit model was more accurate than a continuous model in both the original swine flu and independent COVID-19 datasets.

References

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[2] Elvan Akın and Gülsah Yeni. On exact solutions to epidemic dynamic models. *J Appl Anal Comput*, 10(6):2299–2312, 2020.

[3] European Centre for Disease Prevention and Control. data.europa.eu. Retrieved from `https://data.europa.eu/data/datasets/covid-19-coronavirus-data?locale=en`, 2020. Accessed on 16/3/24.

[4] 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.

[5] C Connell McCluskey. Global stability for an sir epidemic model with delay and nonlinear incidence. *Nonlinear Analysis: Real World Applications*, 11(4):3106–3109, 2010.

[6] NK Vaidya, M Morgan, T Jones, Lapin Miller, S Lapin, and EJ Schwartz. Modelling the epidemic spread of an h1n1 influenza outbreak in a rural university town. *Epidemiology & Infection*, 143(8):1610–1620, 2015.