Covid-19 First Wave SIR Model

Richard Kaufman

ES55 Final Project

Table of Contents

Description

The program analyzes and compares Covid-19 case data from two countries. More specifically, a Susceptible-Infected-Removed (SIR) model is developed for the first wave of the pandemic in each country. The SIR model is a set of three ordinary differential equations that depict the susceptible, infected, and removed (recovered cases and deaths) populations as a function of time. The program finds the optimal infection rate, recovery rate, and scaling factor needed to fit the SIR model to the true active case data.

Inputs

Imported Data: 4 csv files

1. List of countries included in the sample data
    a. Austria
    b. France
    c. Germany
    d. Israel
    e. Italy
    f. Japan
    g. Romania
    h. Russia
    i. South Korea*
    j. Switzerland
2. Cumulative Confirmed Covid-19 cases from 01/22/2020 - 12/13/2020
3. Cumulative Covid-19 deaths from 01/22/2020 - 12/13/2020
4. Cumulative Covid-19 recoveries from 01/22/2020 - 12/13/2020

User Input

- The user has one of two options:
    - Manually type in two countries from the sample data to analyze
    - Request the program to pick two countries from the sample data at random to analyze

Outputs

The program does the following

- Imports Covid-19 Data on confirmed cases, recoveries, and deaths
- Asks the user to manually or randomly pick two countries
- Extracts the data from the countries chosen and calculates active cases
- Determines timespan of first wave by finding the day of the first case and the day of the first local minimum of active cases
    - Calculates the first local minimum of active cases by approximating the derivative using the built in MATLAB gradient() function

*South Korea must be entered into the GUI as “SouthKorea”

- Creates SIR simulation to estimate the daily active cases for a given infection rate, recovery rate, and scaling factor using Euler's method
- Determines the error between the true and simulated active cases using the sum of squared residuals
- MATLAB built-in function fminsearch() determines the infection rate, recovery rate, and scaling factor that minimize the sum of the squared residuals
- Displays the confirmed and active cases, recoveries and deaths over time for both countries
- Displays the active cases for the first wave of both countries along with the best SIR model fit, including text listing the optimal infection rate, recovery rate, and scaling factor that produced this best fit
- Displays a bar graph juxtaposing the infection rates and recovery rates of both countries

Numerical Methods

1. **Ordinary Differential Equations:** Euler's Method and systems of first order ODEs
2. **Differentiation:** forwards, backwards, and centered differentiation using built-in MATLAB function gradient()
3. **Optimization:** three variable optimization using MATLAB built-in function fminsearch()
4. **Random Numbers:** MATLAB built-in function randperm()
5. **Least Squares Regression:** sum of the square of residuals
6. **Graphics**: plots of results and graphical user interface

Validation of Results

In order to ensure the validity of results, the countries analyzed needed to have accurate Covid-19 data. The initial datasets from Johns Hopkins University (JHU CSSE, 2020) included daily data on Covid-19 deaths, recoveries and confirmed cases. While the confirmed case data was robust, the recoveries and deaths suffered from gaps in the data for many countries. This proved to be an issue, as all three variables were needed in order to calculate the daily number of active cases. I posited that obtaining the missing data from other sources could impact the precision of the results, as different datasets may have obtained their information through differing means. To address this issue, for the final project submission I decided to include a subset of sample countries, chosen based on the completeness of the data. All the countries in the sample set have complete data on the confirmed cases, recoveries, and deaths. I additionally ensured that the countries represented a diverse range of Covid-19 responses. The sample set includes both countries that have overcome their first wave and flattened the curve as well as those that never experienced a decline in active cases. Including countries of both responses would demonstrate that the model can fit countries diversely affected by the pandemic.

Through visual inspection, one can see that for many countries, the number of active cases is characterized by multiple phases of growth and decline. These phases represent waves of the epidemic. The SIR model is not equipped to simulate multiple waves of the same disease, as

individuals categorized as removed can no longer reenter the susceptible population. With the Covid-19 virus, those who recover from the disease are still susceptible to future infection. I reviewed literature in order to discover how models accurately capture this discrepancy. Another model additionally suggested by Kermack, W.0., and A.G. Mckendrick. (1927), the same researchers who developed the groupwork for the SIR model, was a Susceptible-Infected-Susceptible model. Here, the infected population would be moved back to the susceptible population after overcoming the disease, implying that no immunity is obtained. This also does not accurately reflect Covid-19 however, as there is a period when a previously infected person does have antibodies and thus immunity (Iyer, Anita S., et al., 2020). While other models address this temporary immunity period, they are outside the scope of this course and project. An attempt to address this shortcoming of the SIR model with respect to Covid-19 was Cooper, Ian, et al. (2020). The SIR model assumes homogeneity among the susceptible and infected population. With Covid-19, epicenters may arise independently from these susceptible populations due to travel and globalization. They recommended approaching this by adding a surge period when the susceptible population jumps. This jump in susceptible individuals represents a new subset of the population getting exposed to the virus. A shortcoming with this method, however, is that Cooper, Ian, et al. (2020) assigned these surge periods based on visual inspection of the accepted active case data. Given the goals of my project, this would not be a feasible task to undertake. Based on this background research, I decided the best solution to maintain the accuracy of my model would be to only model the first wave. I defined the first wave as the time between the first reported positive case and when the active cases reached their first local minimum. The first wave would be considered ongoing for countries that did not successfully "flatten the curve". Here, the analysis would end before any significant surge in susceptible individuals would facilitate a second wave. Additionally, this decision would allow me to make the assumption that the recovered population would not contract the virus a second time.

Now that the data and methods have been justified, I will evaluate the validity of my results. When creating the SIR model to fit the active case data, I made several key assumptions. Starting, given the short time frames of at most 10 months, any births and deaths not related to Covid-19 are negligible. Thus, the population under analysis, N, is constant. Another assumption was made regarding the maximum possible number of infected individuals. With the model I am using (see appendix 3), there are four unknown parameters. To avoid an excessive number of iterations when finding the optimal SIR model, I make the assumption that the maximum portion of infected people is 1, meaning up to N people can be infected with the model I make. In addition to assumptions, I also addressed the concern of compounding error with differentiation. The active case data did not always follow a smooth pattern and would sometimes move in inconsistent directions. Differentiating this data could compound the existing error. To address this, I took the 15-day rolling average of the active case data before taking the derivative. I found this method to be effective, as following this adjustment the program accurately estimated the end of the first wave.

To ensure the accuracy of my fit, I used a step size of one day: the highest resolution available. Looking at sample figures 4 and 6 in appendix A, one can see that the SIR fits accurately match the true number of active cases. From the similarity between these two lines, I

can conclude that the parameters utilized to create the SIR model accurately reflect the spread of active Covid-19 cases in the respective country. To further validate this, I compared my results for South Korea and Italy from Cooper, Ian, et al. (2020)

Country	South Korea			Italy		
Source	My program	Cooper, Ian, et al., 2020	Percent Error	My program	Cooper, Ian, et al., 2020	Percent Error
Infection Rate	.284	.400	40.8%	.212	.180	17.8%
Recovery Rate	.037	.035	5.7%	.022	.037	40.5%

From the table comparison one can see that the infection and recovery rates that I calculated were of the same magnitude of those calculated by Cooper, Ian, et al. (2020), however overall deviated from their assessment. I attribute the discrepancy in our values to the difference in methodology utilized. I assigned the maximum infected population while their paper chose it via visual inspection. Additionally, I only analyzed the first wave, while their parameter values were calculated using surges and a longer time span. Finally, the discrepancies may be due to the different datasets utilized by their study and my project.

References

Abou-Ismail, Anas. “Compartmental Models of the COVID-19 Pandemic for Physicians and Physician-Scientists.” *SN Comprehensive Clinical Medicine*, vol. 2, no. 7, Springer Science and Business Media LLC, July 2020, pp. 852–58, doi:10.1007/s42399-020-00330-z.

Chapra, S.C. 2007. Applied Numerical Methods with MATLAB for Engineers and Scientists. McGraw-Hill, New York.

Cooper, Ian, et al. “A SIR Model Assumption for the Spread of COVID-19 in Different Communities.” *Chaos, Solitons and Fractals*, vol. 139, Elsevier Ltd, Oct. 2020, p. 110057, doi:10.1016/j.chaos.2020.110057.

GitHub - CSSEGISandData/COVID-19: Novel Coronavirus (COVID-19) Cases, Provided by JHU CSSE. https://github.com/CSSEGISandData/COVID-19. Accessed 17 Dec. 2020.

Iyer, Anita S., et al. “Persistence and Decay of Human Antibody Responses to the Receptor Binding Domain of SARS-CoV-2 Spike Protein in COVID-19 Patients.” *Science Immunology*, vol. 5, no. 52, American Association for the Advancement of Science, Oct. 2020, doi:10.1126/sciimmunol.abe0367.

Kermack, W.0., and A.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, vol. 115, no. 772, The Royal Society, Aug. 1927, pp. 700–21, doi:10.1098/rspa.1927.0118

Appendix A: Sample: Output:

Figure 1. Initial GUI prompt

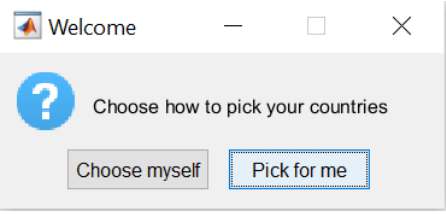
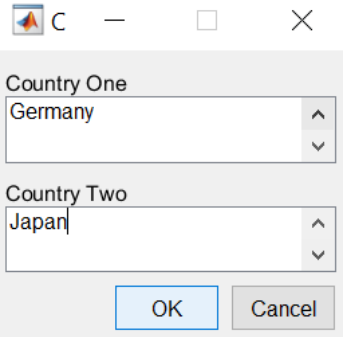
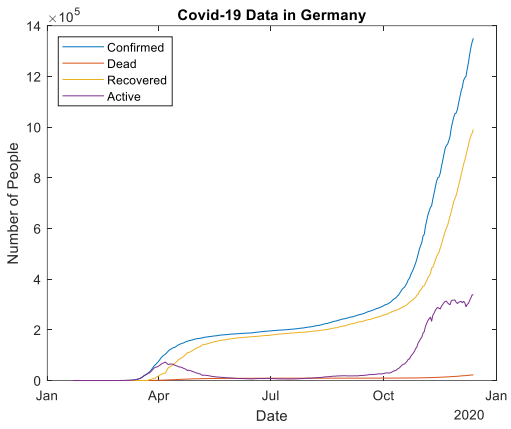
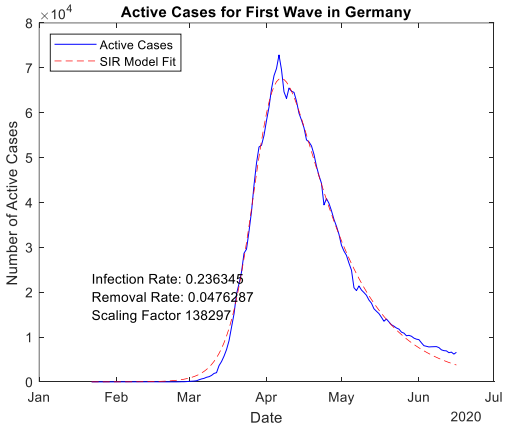


Figure 2. Germany and Japan selected as countries to analyze

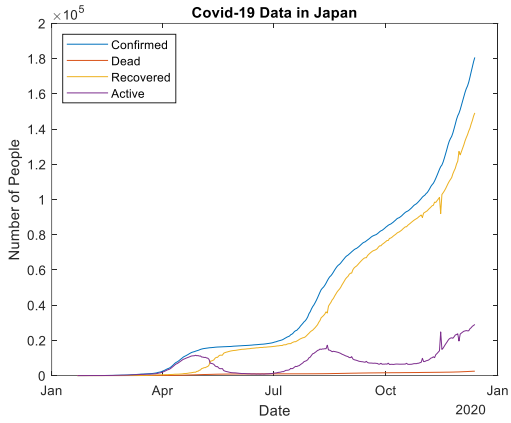


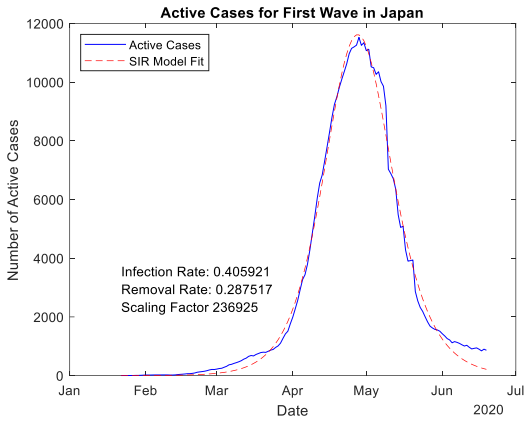
Figures 3&4. Germany analysis



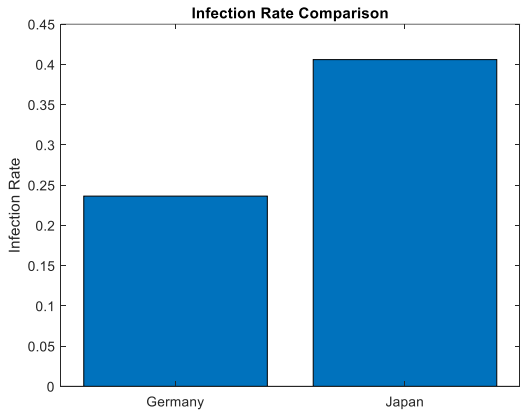


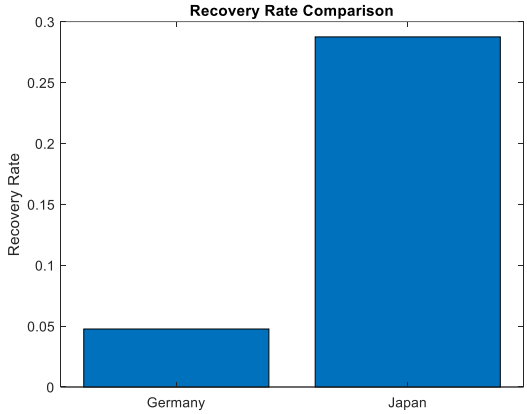
Figures 5&6. Japan analysis





Figures 7&8. Comparison of infection and recovery rates





Appendix B: Code

Kaufman_Project2020.m (script)

```
%Richard Kaufman 12/18/20 ES55
%Kaufman_project_2020 imports Covid-19 data and uses an SIR model to
%estimate the rate of infection and rate of recovery for two countries

%Imported Covid-19 Data
warning('OFF', 'MATLAB:table:ModifiedAndSavedVarnames')
confirmed = readtable('Covid-19 Confirmed Cases Sample.csv');
deaths = readtable('Covid-19 Deaths Sample.csv');
recovered = readtable('Covid-19 Recovered Sample.csv');
countryList = readtable('Country List.csv');

%GUI to pick the two countries for analysis
stop = 0;
answer = questdlg('Choose how to pick your
 countries','Welcome','Choose myself','Pick for me','Pick for me');
while stop ~= 2
    if answer == "Pick for me"
        indices = randperm(length(countryList.('Country')),2);
        country1 = countryList.('Country')(indices(1));
        country2 = countryList.('Country')(indices(2));
        country1 = country1{1,:};
        country2 = country2{1,:};
        break
    end
    stop = 0;
    prompt = {'Country One', 'Country Two'};
    inputTitle = 'Compare Two Countries';

    country = inputdlg(prompt, inputTitle,2);
    country1 = country{1,1};
    country2 = country{2,1};
    for i = [1:length(countryList.('Country'))]
        tempCountry = countryList.('Country')(i);
        if convertCharsToStrings(country1) == ...
                convertCharsToStrings(tempCountry{1,:}) ||...
                convertCharsToStrings(country2) == ...
                convertCharsToStrings(tempCountry{1,:})
            stop = stop +1;
        end
    end
    if stop ~=2
        answer = questdlg('One or more countries entered are not in
 the database','Error','Retry','Pick for me','Pick for me');
    end
end

dates = confirmed.('Date');

%creating arrays from table data for country 1
[confirmed1, recovered1, deaths1, trueActive] = ...
getCountryData(confirmed, recovered,deaths, country1);

%Finding first wave for country 1
```

```
[start, finish] = getPeriod(trueActive);

%Finding the best SIR model and parameters for country 1
h = 1; %Step size of one day
param0 = [.2,.05, 100000]; %Initial guess
[infected,optimalParams1] = getOptimalModel(finish, start, h,...
    param0, trueActive);

plotTitle = sprintf('Covid-19 Data in %s', country1);
figure(1)
plot(dates,confirmed1)
hold on
plot(dates, deaths1)
plot(dates, recovered1)
plot(dates, trueActive)
legend('Confirmed', 'Dead', 'Recovered', 'Active', 'Location', 'northwest')
xlabel('Date')
ylabel('Number of People')
title(plotTitle)
hold off

plotTitle = sprintf('Active Cases for First Wave in %s', country1);
results = sprintf('Infection Rate: %g\nRemoval Rate: %g
\nScaling Factor %g', optimalParams1(1), optimalParams1(2),
 round(optimalParams1(3),0));
figure(2)
```

```
plot(dates(start:finish),trueActive(start:finish), 'b-')
hold on
plot(dates(start:finish),infected, 'r--')
legend('Active Cases','SIR Model Fit','Location', 'northwest')
xlabel('Date')
ylabel('Number of Active Cases')
title(plotTitle)
text(dates(start),mean(trueActive(start:finish)),results)
hold off

%Extracting arrays from tables for country 2
[confirmed2, recovered2, deaths2, trueActive] = ...
getCountryData(confirmed, recovered,deaths, country2);

%finding first wave date indices for country 2
[start, finish] = getPeriod(trueActive);

%Finding the best SIR model and parameters for country 2
h = 1; %Step size of one day
param0 = [.1,.05, 100000]; %Initial Guess
[infected,optimalParams2] = getOptimalModel(finish, start, h,...
    param0, trueActive);

plotTitle = sprintf('Covid-19 Data in %s', country2);
figure(3)
plot(dates,confirmed2)
hold on
plot(dates, deaths2)
```

```
plot(dates, recovered2)
plot(dates, trueActive)
legend('Confirmed', 'Dead', 'Recovered', 'Active', 'Location', 'northwest')
xlabel('Date')
ylabel('Number of People')
title(plotTitle)
hold off

plotTitle = sprintf('Active Cases for First Wave in %s', country2);
results = sprintf('Infection Rate: %g\nRemoval Rate: %g
\nScaling Factor %g', optimalParams2(1), optimalParams2(2),
 round(optimalParams2(3),0));
figure(4)
plot(dates(start:finish),trueActive(start:finish), 'b-')
hold on
plot(dates(start:finish),infected, 'r--')
legend('Active Cases','SIR Model Fit','Location', 'northwest')
xlabel('Date')
ylabel('Number of Active Cases')
title(plotTitle)
text(dates(start),mean(trueActive(start:finish)),results)
hold off

%Comparing country 1 and country 2
figure(5)
X = categorical({country1, country2});
Y = [optimalParams1(1) optimalParams2(1)];
bar(X,Y)
title('Infection Rate Comparison')
ylabel('Infection Rate')

figure(6)
X = categorical({country1, country2});
Y = [optimalParams1(2) optimalParams2(2)];
bar(X,Y)
title('Recovery Rate Comparison')
ylabel('Recovery Rate')
```

getCountryData.m (function)

```
function [confirmed, recovered, deaths, active] = ...
getCountryData(confirmed, recovered,deaths, country)
%getCountryData: calculate active cases
%function [confirmed, recovered, deaths, active] =
 getCountryData(confirmed, recovered,deaths, country)
%Richard Kaufman 12/18/20 ES55
%   converts table data to arrays for a specified country and
 calculates active cases
%Inputs
%   confirmed: Covid-19 daily confirmed cases
%   recovered: Covid-19 daily recovery data
%   deaths: Covid-19 daily death data
%   country: the country to retrieve data for
%Output:
%   confirmed: confirmed cases for the specified country
%   recovered: recovered cases for the specified country
%   deaths:deaths for the specified country
%   active:active cases for the specifed country

confirmed = confirmed.(country);
recovered = recovered.(country);
deaths = deaths.(country);
active = confirmed - recovered - deaths;

end
```

getPeriod.m (function)

```
function [start,finish] = getPeriod(activeCases)
%getPeriod: time period first wave
%function [start,finish] = getPeriod(activeCases)
%Richard Kaufman 12/18/20 ES55
%   calculates the indices of the start and end of the first wave
%Inputs
%   activeCases: the daily active case data
%Output:
%   start: index of the first day of the first wave
%   finish: index of the last day of the first wave

%calculate derivative of active case data
smoothActive = movmean(activeCases,15);
smoothDiff = gradient(smoothActive);

%finds first local minimum
finish = length(activeCases);
for i = [50:length(smoothDiff)-1]
    if smoothDiff(i)*smoothDiff(i+1) < 0 && smoothDiff(i)...
        <smoothDiff(i+1) || smoothDiff(i)==0
        finish = i;
        break
    end
end

%finds date of first Covid-19 case
start = 1;
while activeCases(i) == 0
    start = start+1;
end

end
```

calcInfected.m (function)

```
function infected = calcInfected(param,finish,start,h)
%calcInfected: SIR solution Euler's Method
%the active cases over a time period and given SIR parameters
%infected = calcInfected(param,finish,start,h)
%Richard Kaufman 12/18/20 ES55
%   uses the SIR model and Euler's method to estimate
%the active cases over a time period and given SIR parameters
%Inputs
%   param: length 3 array containing infection rate, recovery rate,
%   and scaling factor respectively
%   start: index of the first day of first wave
%   finish: index of the last day of first wave
%   h: step size
%Output:
%   infected: time dependent active case data from SIR model

    beta = param(1);
    K = param(2);

    %initializing active cases
    infected = zeros((finish - start +1)/h,1);
    infected(1) = 1/param(3);

    %initializing susceptible population
    susceptible = zeros((finish - start+1)/h,1);
    susceptible(1) = 1 - infected(1);


    for i = [2:length(susceptible)]

        %Calculate change in variables from previous iteration
        drdt = infected(i-1)*K;
        didt = beta*infected(i-1)*susceptible(i-1) - drdt;
        dsdt = beta*infected(i-1)*susceptible(i-1);

        %Apply Eurler's Method
        infected(i) = infected(i-1)+didt*h;
        susceptible(i) = susceptible(i-1) - dsdt*h;
    end

    %Scaling the active case data
    infected = infected*param(3);
end
```

getError.m (function)

```
function Sr = getError(param, finish,start,h, trueActive)
%getError: active case sum of the squared residuals
%function Sr = getError(param, finish,start,h, trueActive)
%Richard Kaufman 12/18/20 ES55
%   computes estimate of the active cases and compares this to the
%accepted active case data
%Inputs
%   param: length 3 array containing infection rate, recovery rate,
%   and scaling factor respectively
%   finish: index of the last day of first wave
%   start: index of the first day of first wave
%   h:step size
%   trueActive: accepted active case data
%Output:
%   Sr: Sum of the squared residuals between the accepted and estimate
%   values

infected = calcInfected(param, finish, start, h);

Sr = sum((trueActive(start:finish) - infected).^2);

end
```

getOptimalModel.m (function)

```
function [infected, optimalParam] = getOptimalModel(finish,...
    start, h,param0, trueActive)
%getOptimalModel: optimization SIR parameters
%function [infected, optimalParam, error] =
 getOptimalModel(finish,start, h,param0, trueActive)
%Richard Kaufman 12/18/20 ES55
%   finds the optimal infection rate, recovery rate, and scaling
 factor
%Inputs
%   finish: index of the last day of first wave
%   start: index of the first day of first wave
%   h:step size
%   param0: initial guess for optimization
%   trueActive: accepted active case data
%Output:
%   infected: the optimal active case data from the SIR model
%   optimalParam: optimal infection rate, recovery rate, and scaling
 factor

%finds optimal paramters
errorFunction = @(param) getError(param,finish, start,h, trueActive);
options = optimset('MaxFunEvals', 3000, 'MaxIter',3000);
optimalParam = fminsearch(errorFunction, param0, options);

%uses parameters to calculate final estimate for active cases
infected = calcInfected(optimalParam, ...
finish,start,h);

end
```

Appendix 3: Mathematics Details

Given the raw data on Covid-19 confirmed cases, recoveries, and deaths, we can calculate the number of active cases using the formula:

Active Cases = Confirmed Cases – Recoveries – Deaths

Following the calculation of active cases, I find the start and end dates of the first wave. I do this by using the MATLAB built-in function gradient(), which takes the forward difference for the first point, a centered difference for the intermediate points, and a backwards difference for the last point (Chapra, 2007). The local minimum occurs where the derivative is equal to 0 and crosses from a negative to positive value.

I next employ the Susceptible-Infected-Removed model, first developed by Kermack, W.0., and A.G. Mckendrick (1927) and described with respect to Covid-19 by Abou-Ismail, Anas (2020). In this model, the constant total population, N, consists of the time-dependent susceptible, infected, and removed population:

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

In the equation above, S(t) represents the number of people who can come into contact with an infected individual within the population. I(t) represents the active cases, and R(t) represents the sum of the recoveries and epidemic-related deaths. With SIR analyses, they are often normalized to N, and thus this equation is rearranged to yield:

$$\frac{\boldsymbol{S(t)}}{\boldsymbol{N}} + \frac{\boldsymbol{I(t)}}{\boldsymbol{N}} + \frac{\boldsymbol{R(t)}}{\boldsymbol{N}} = \mathbf{1}$$

The susceptible, infected, and removed parameters normalized to N create new parameters, represented by their lower-case counterparts:

$$\frac{\boldsymbol{S(t)}}{\boldsymbol{N}} = \mathbf{s(t)} \quad \frac{\boldsymbol{I(t)}}{\boldsymbol{N}} = \mathbf{i(t)} \quad \frac{\boldsymbol{R(t)}}{\boldsymbol{N}} = \mathbf{r(t)}$$

$$\mathbf{s(t) + i(t) + r(t) = 1}$$

With these definitions, we can now examine the three governing differential equations of the SIR model. These equations depict the changes in the susceptible, infected, and removed population with respect to time, and are as follows:

$$\frac{\boldsymbol{ds(t)}}{\boldsymbol{dt}} = \mathbf{-\beta s(t)i(t)}$$

$$\frac{\boldsymbol{di(t)}}{\boldsymbol{dt}} = \mathbf{\beta s(t)i(t) - \kappa i(t)}$$

$$\frac{\boldsymbol{dr(t)}}{\boldsymbol{dt}} = \mathbf{- \kappa i(t)}$$

Here, β represents the rate of infection and κ represents the rate of recovery. These parameters can be adjusted to alter the shape of the SIR model. The three differential equations are all ordinary and first order. Additionally, the system has three ODEs and three unknowns s(t), i(t), and r(t), meaning system can be solved. We approach this system by using Euler's method (Chapra, 2007) to approximate the values of s(t), i(t), and r(t) at each time increment, separated by time step h.

$$\mathbf{s(t+h) = s(t) + \frac{ds(t)}{dt}h}$$

$$\mathbf{i(t+h) = i(t) + \frac{di(t)}{dt}h}$$

$$\mathbf{r(t+h) = r(t) + \frac{dr(t)}{dt}h}$$

Using these expressions and Euler's method, we only additionally need the initial conditions of the system in order to find s(t), i(t), and r(t) at each timestep. To approach this, I employed the method used in Cooper, Ian, et al. (2020). Here, the initial fraction of infected individuals is defined as:

$$\mathbf{i(0) = \frac{imax}{f}}$$

In this formula i_{max} represents the maximum infections, assumed to be 1, and *f* represents the scaling factor, which is solved iteratively. The final time-dependent infection data will be multiplied by the scaling factor to match the true active case values. We now see the other two initial conditions:

$$\mathbf{s(0) = 1 - i(0)}$$

$$\mathbf{r(0) = 0}$$

With these initial conditions, we claim that at the start of the epidemic, zero individuals are recovered. Most of the population is susceptible, save the small portion that start out in the model infected.

We see from the formulas above that three parameters necessary for solving the three ODEs are unknown: β, κ, and *f*. These parameters can be optimized so that the SIR model will accurately fit the accepted active case data. To accomplish this, I utilized the MATLAB built-in function fminsearch(), which based on the Nelder-Mead method (Chapra, 2007), finds the minimum of a multidimensional function. Here, the value minimized was the error between the SIR and accepted active cases and the three dimensions to be adjusted were β, κ, and *f*. The values that produced the SIR infected curve with the lowest error were designated as the epidemic parameters assigned to the country analyzed.