University of California, Santa Barbara

MATH 104B

FINAL PROJECT

An Application of Numerical Analysis to the COVID-19 Spread in Hubei, China

Authors:
Emily Lu
Alberto Beltran
Matthew Peters

Supervisor:
Professor Paul Atzberger

March 4th, 2020

Abstract

The background area being discussed is the COVID-19 disease and specifically modeling the number of people infected as a whole within the Hubei Province. Using data gathered from the World Health Organization (WHO) and taking into account the number of cases reported, the contagion rate, and the total population, we are able to model the number of individuals infected with the COVID-19 disease compared to the reported number of confirmed cases. In this project, we aim to determine the number of individuals in the Hubei Province infected with the disease compared to the number of reported cases, which allows us to determine the reproductive ratio of the disease. The methods used to solve the question at hand will be Numerical Linear Matrix operations, and approximating the Adam Moulton methods using the exponential growth formula.

Contents

1 Introduction

The objectives are to understand better the mathematical dynamics of an infected population when an outbreak occurs. Another goal is to use mathematical modeling to examine and to analyze the viral dynamics of the COVID-19 disease. To model this outbreak, the systems of differential equations are used. We decided to model the COVID-19 disease in the Hubei Province in February 2020 and compare the disease spread using an SIR model.

The purposes of this research are:

1. To apply SIR model to predict the outbreaks of COVID-19 virus.
2. To determine the effect of the initial number of infected individuals of the population.

2 Modeling the Spread of Infected

The COVID-19 disease has been present for 1 month in the Hubei Province. According to reports, the disease has an average natural lifespan of 30 days. The contagion rate for the first 10 days is 1.9, the contagion rate for the next 10 days is 2.6, and the contagion rate for the last ten days is 3.6. We reported the total number of confirmed cases at 10 day intervals for a month (30 days) in an effort to compute the ratio of infected (with respect to the total population). According to WHO, the population of Hubei is approximately 59 million people. Thus, we took 59 million as the fixed population of Hubei and gathered the daily number of new cases from February 1st, 2020 to March 2nd, 2020. Using these figures, we calculated the average number of cases reported within the first 10 days to be 2247.8, the average number of cases within the next 10 days to be 1325.7, and the average number of cases within the last ten days to be 464.5. Afterwards, we divided each values by the total population and set it as the first row of matrix A. Using these informations, we generated a rough estimate of the actual number of people infected by taking into account the spread factor of the disease by using the following steps below:

Step 1: Define matrix A.

A matrix $A = [a_{ij}]$ can be used to model the COVID-19 disease transmission rate (based on the number of cases reported, the contagion rate, and the total population after adjusting for fatalities) by letting a_{ij} denote the probability that an infected person from the ith time period transmitted the COVID-19 disease to a susceptible person in the jth time period. With this in mind, we could solve for the first row of matrix A which contains the averaged contagion rates of each time interval based on actual number of cases.

According to WHO based on the situation reports in February 2020, the "incubation period" or the time frame between catching COVID-19 and beginning to have symptoms of the disease could range from 2-10 days. Thus, we selected 10 days to the possible time period an infected person could have spreaded the disease to others since we were working with data over a 30-days period. With the first 10 days being the possible incubation period, we solved for the following:

- a_{21} which is determined by adding the confirmed cases from 1-10 days and 10-20 days, then multipying by the contagion rate 2.6 followed by dividing the respective total population and
- a_{32} which is determined by adding the confirmed cases from 10-20 days and 20-30 days, then multipying by the contagion rate 3.1 followed by dividing the respective total population.

$$A = \begin{bmatrix} 0.00038098 & 0.0002247 & 0.00007873 \\ 0.0015747801 & 0 & 0 \\ 0 & 0.0009406356 & 0 \end{bmatrix}$$

Step 2: Define matrix x and solve for vector b.

Vector x is population per time period (10 days) which was found by re-evaluating each row following the first by taking into account the number of deaths from the prior 10 days.

Vector b is average number of reported / population per 10 days. Therefore, vector b has the following rows:

$$b_{11} \; : \; \text{total number of confirmed cases over 30 days.} \tag{1}$$

$$b_{21} \; : \; \text{total number of infected after 20 days.} \tag{2}$$

$$b_{31} \; : \; \text{total number of infected after 30 days.} \tag{3}$$

Our goal was then to solve for vector b through the matrix multiplication of Ax.

$$\begin{bmatrix} 0.00038098 & 0.0002247 & 0.00007873 \\ 0.0015747801 & 0 & 0 \\ 0 & 0.0009406356 & 0 \end{bmatrix} \begin{bmatrix} 59000000 \\ 58999350 \\ 58998618 \end{bmatrix} = \begin{bmatrix} 40379.93514014 \\ 92912.0259 \\ 55496.88898686 \end{bmatrix} \tag{4}$$

From the results of vector b, we could see that number of people infected drastically increased from the first and second time interval, but decreased from the second to third time interval. This is plausible because the Hubei Province started their quarantine process in late January and thus, decreased the contagion rate of the COVID-19 disease.

Step 3: Use the exponential growth formula to measure the rate of spread of the disease in 1 month in advance.

Since the COVID-19 disease is an airborne virus, we applied the explonential growth formula, using $b_{31} = 55496$ as the initial number of infected since b_{31} is denoted as the most "current" in our case, to approximate number of infected people in the future. We modeled this as

$$y = 55496e^{2t}$$

and we adapted this solution to an initial value problem,

$$f(t, y) = 2ye^{2t}.$$

This initial value problem $f(t, y)$ is used to find our exponential growth approximations through the Runge-Kutta and Adam Moulton methods.

	Runge Kutta 4-Step	AM 2-Step	AM 3-Step	AM 4-Step	Actual
0	55496	55496	55496	55496	55496
0.05	61651	61651	61651	61651	61333
0.1	69249	69249	69249	69249	67783
0.15	78741	74320	78741	78741	74912
0.2	90752	80304	85678	90752	82790
0.25	106169	87454	93935	100068	91497

	Runge Kutta 4-Step	AM 2-Step	AM 3-Step	AM 4-Step	Actual
0.3	126271	96068	103975	111248	101120
0.35	152942	106537	116280	125098	111755
0.4	189018	119380	131516	142340	123509
0.45	238866	135305	150599	164075	136498
0.5	309380	155277	174801	191841	150854
0.55	411761	180643	205920	227831	166719
0.6	564744	213305	246538	275225	184253
0.65	800727	255996	300431	338728	203631
0.7	1177761	312715	373227	425438	225047
0.75	1804025	389429	473491	546304	248716
0.8	2889924	495215	614547	718605	274873
0.85	4864357	644192	817618	970298	303782
0.9	8647783	858837	1117364	1347830	335731
0.95	16330432	1175828	1571944	1930606	371040
1	32962874	1656629	2281796	2858551	410063

3 Modeling the COVID-19 as a SIR Model

We used the SIR model to illustrate the transfer of the epidemic disease through the interaction of the following three different variables:

> S = Number of people that are susceptible to COVID-19 I = Number of people infected with COVID-19 R = Number of people recovered from COVID-19 with total immunity

For simplicity, we assumed a fixed population of N people, where

$$N = S + I + R;$$

in other words, there will be no births or deaths taken into account.

The SIR model uses 3 differential equations:

$$\frac{dS}{dT} = -\beta IS; \; \frac{dI}{dT} = \beta IS - \gamma I; \; \frac{dR}{dt} = \gamma I \tag{5}$$

where $S(t)$ is the number of susceptible people in the population at time t, $I(t)$ is the number of infectious people at time t, $R(t)$ is the number of recovered people at time t, β is the transmission rate, γ represents the recovery rate. Note: in our SIR model, we set $t = 0$ for the start of February 2020.

Now given our research on the COVID-19 outbreak in the Hubei Province from February 1st, 2020 to March 2nd, 2020 through WHO, we assigned the parameters with the following values:

> $N = 59000000$ which is the total population of Hubei, $I = 40380$ which is the number of people infected in the month of February,

and solved for the rest.

The number of people dead in the month of February is 1382. However, seeing as R includes the number of people who have received permanent immunity, this could be simplified to include those who have died as they'll have permanent immunity afterwards and those who have recovered with permanent immunity. Therefore, number of people recovered could be defined as $R = 1382 + (0.6 \times 40380) = 25,610$.

The duration of the disease ranges from 1 to 30 days [10]; therefore we could roughly estimate the duration of the disease at the midpoint, i.e.15 (approx.) days:

$D = 15$; $\gamma = 1/15 = 0.067$.

According to WHO, the mortality rate of COVID-19 is 0.4 [3] and the total population of the Hubei Province after accounting for deaths is 58,998,618. Therefore, β (the rate of infection) $= 0.4/58,998,618 = 6.7798 \times 10^{-9}$, $S = 58932628$, and $N = 58,998,618$.

In order to use the SIR model to predict the evolution of the disease, it would be helpful if we could solve the system of differential equations. Therefore, we used numerical approaches such as the fourth order Runge-Kutta method to extract the solution.

	Susceptible (S)	Infectious (I)	Recovered (R)	Population (N)
0	58932628	40380	25610	58998618
1	58913489	56325	28804	58998618
2	58886806	78553	33259	58998618
3	58849616	109531	39471	58998618
4	58797808	152679	48131	58998618
5	58725680	212737	60201	58998618
6	58625358	296247	77013	58998618
7	58485996	412206	100416	58998618
8	58292742	572915	132961	58998618
9	58025409	795049	178160	58998618
10	57656846	1100955	240817	58998618
11	57151076	1520087	327454	58998618
12	56461423	2090360	446835	58998618
13	55529140	2858923	610554	58998618
14	54283491	3881478	833648	58998618
15	52644802	5218754	1135063	58998618
16	50532445	6928420	1537753	58998618
17	47879491	9051083	2068044	58998618
18	44653900	11590849	2753870	58998618
19	40882122	14494857	3621639	58998618
20	36665858	17640811	4691949	58998618
21	32180491	20843096	5975030	58998618
22	27649046	23882181	7467391	58998618
23	23298099	26549510	9151009	58998618
24	19313186	28689887	10995545	58998618
25	15811305	30224512	12962801	58998618
26	12837123	31149603	15011892	58998618
27	10377242	31517688	17103688	58998618
28	8381680	31413184	19203754	58998618
29	6783706	30931170	21283742	58998618
30	5513972	30163069	23321577	58998618

4 Initial Value Problem Analysis

The proposed differential equations with,

$$0 \leq \beta \leq 6.778 * 10^{-9}, \quad 0 \leq \gamma \leq .067$$

are:

$$\frac{dS}{dt} = -\beta IS \frac{dI}{dt} = \beta IS - \gamma I \frac{dR}{dt} = \gamma I. \tag{6}$$

To illustrate the robustness of these differential equations, we will aim to satisfy a Lipschitz condition and show their well-possedness.

Define

$$D = [(t, S, I, R)| \quad 0 \leq t \leq 30, \quad and \quad -\infty < S, I, R < \infty]$$

Equation (1):

$$\frac{dS}{dt} = -\beta IS = f(t, S, I)$$

And so

$$|f(t, S_1, I_1) - f(t, S_2, I_2)| = \beta|I_2S_2 - I_1S_2| \leq 6.778 * 10^{-9}|I_2S_2 - I_1S_2|.$$

The equation then satisfies a Lipschitz condition with constant

$$L = 6.778 * 10^{-9}.$$

In addition, $f(t, S, I)$ is clearly defined everywhere on D. So by Theorem 5.6, we know that (1) is well-posed.

Equation (2):

$$\frac{dI}{dt} = \beta IS - \gamma I = f(t, S, I)$$

And so

$$|f(t, S_1, I_1) - f(t, S_2, I_2)| = |\beta(I_1S_1 - I_2S_2) + \gamma(I_2 - I_1)| \leq \beta|I_1S_1 - I_2S_2| + \gamma|I_2 - I_1|$$

Now let

$$L = max(|\beta|, |\gamma|).$$

Hence,

$$f(t, S_1, I_1) - f(t, S_2, I_2) \leq L(|I_1S_1 - I_2S_2| + |I_2 - I_1|).$$

And (2) then satisfies a Lipschitz condition with constant

$$L = \gamma = .067.$$

In addition, $f(t, S, I)$ is clearly defined everywhere on D. So by Thm. 5.6 we know that (2) is well-posed.

Equation (3):

$$\frac{dR}{dt} = \gamma I = f(t, S, I)$$

And so

$$|f(t, S_1, I_1) - f(t, S_2, I_2)| = \gamma|I_1 - I_2| \leq .067|I_1 - I_2|.$$

The equation then satisfies a Lipschitz condition with constant

$$L = .067.$$

In addition, $f(t, S, I)$ is clearly defined everywhere on D. So by Thereom 5.6 from the *Numerical Analysis 10 Ed.*, we know that (3) is well-posed.

5 Concluding Remarks

In this project, our objective was to mathematically model the number of people infected with the novel COVID-19 disease. More specfically, we wanted to first simulate the number of people infected through a mathematical model compared to the reported number of reported cases from WHO's February 1st, 2020 to March 2nd, 2020 situation reports. To do so, we utilized the Numerical Linear Matrix operations and then, the Adam Moulton methods. We found the next month results through the Adams Moulton methods to significantly vary from the supposed actual reported cases. A plausible explanation for this is that these numerical methods do not take into account of the quarantine effect on the contagion rate of the disease. With in this mind, we then decided to model the problem as SIR model instead. We started by defining an SIR model representation of the COVID-19 by finding the initial value problem. After, simulating our SIR model using exponential growth and the Runge-Kutta 4th order method, we found our number of infected people not too far off from our Runge-Kutta values we obtained in the first part of our project. Since the initial value problem we defined for our SIR model which was obtained by observing a physical phenomena generally only approximate the true situation, we wanted to know whether a small change in the statement of our problem would also introduce correspondingly small changes. To do so, we tested for the well-posedness of our problem and found that it is well-posed. Thus, we concluded that our SIR model is a valid representation of the COVID-19 disease transmission spread. However, given the nature of our model, it is not a perfect representation as it does not take into account the number of deaths, the immunity rate, and other control factors we may have been shortsighted with. Overall, it was interesting to learn how numerical analysis could be used to mathematically model the possible contagion rates if the government of China did not step in to quarantine their people in the Hubei Province.

Appendix: Additional Python Code

```python
import numpy as np
import pandas as pd

# Matrix model of the proportion of people infected as a whole
A = [[0.00038098, 0.0002247, 0.00007873],
   [0.0015747801, 0, 0],
   [0, 0.0009406356, 0]]

# Matrix model of the population per time period
x = [[59000000],
   [58999350],
   [58998618]]

# Matrix model of the total population infected per time period
b = np.matmul(A, x)

# Time sequence array of size N
def t_i():
    t = np.zeros(n+1)
    t[0] = a
    for i in range(n):
        t[i+1] = t[i] + h
    return t

# Runge Kutta 4-Step method; used to estimate the first
# few values for the Adams Moulton methods
def rk4(n):
    w = np.zeros(n+1)
    w[0] = alpha
    for i in range(n):
        k1 = h*f(t[i], w[i])
        k2 = h*f(t[i] + h/2, w[i] + k1/2)
        k3 = h*f(t[i] + h/2, w[i] + k2/2)
        k4 = h*f(t[i] + h, w[i] + k3)
        w[i+1] = w[i] + 1/6*(k1 + 2*k2 + 2*k3 + k4)
    return w

# Adams Moulton 2-Step Method
def adamsMoult_2(y):
    w = np.zeros(n+1)
    w[0], w[1], w[2] = y[0], y[1], y[2]
    for i in range(2, n):
        w[i+1] = w[i] + h/12*(5*f(t[i+1], w[i+1]) + 8*f(t[i], w[i])
                 - f(t[i-1], w[i-1]))
    return w
```

```python
# Adams Moulton 3-Step Method
def adamsMoult_3(y):
    w = np.zeros(n+1)
    w[0], w[1], w[2], w[3] = y[0], y[1], y[2], y[3]
    for i in range(3, n):
        w[i+1] = w[i] + h/24*(9*f(t[i+1], w[i+1]) + 19*f(t[i], w[i])
               - 5*f(t[i-1], w[i-1]) + f(t[i-2], w[i-2]))
    return w

# Adams Moulton 3-Step Method
def adamsMoult_4(y):
    w = np.zeros(n+1)
    w[0], w[1], w[2], w[3], w[4] = y[0], y[1], y[2], y[3], y[4]
    for i in range(4, n):
        w[i+1] = w[i] + h/720*(251*f(t[i+1], w[i+1])
               + 646*f(t[i], w[i]) - 264*f(t[i-1], w[i-1])
               + 106*f(t[i-2], w[i-2]) - 19*f(t[i-3], w[i-3]))
    return w

# Iterative function that solves the function y
# at each value in the time array t
def actual(n):
    solutions = []
    for i in range(n+1):
        solutions.append(y(t[i]))
    return solutions

# Initial value problem
def f(t,y):
    return (2*y*np.e**(2*t))

# Solution to the initial value problem above
def y(t):
    return(55496*np.e**(2*t))

alpha = 55496; h = 0.05; a = 0; n = 20; t = t_i()

data = {'Time':t, 'Runge Kutta 4-Step':rk4(n),
        'Adams Moulton 2-Step':adamsMoult_2(rk4(2)),
        'Adams Moulton 3-Step':adamsMoult_3(rk4(3)),
        'Adams Moulton 4-Step':adamsMoult_4(rk4(4)),
       'Actual':actual(n)}

table = pd.DataFrame(data).round()

# SIR model version of the Runge Kutta 4-Step method
def rk4_SIR(n):
    S = np.zeros(n+1); I = np.zeros(n+1); R = np.zeros(n+1)
    S[0] = alpha0; I[0] =  alpha1; R[0] = alpha2

    dSdt = lambda S, I: -beta*S*I
    dIdt = lambda S, I: beta*S*I - gamma*I
    dRdt = lambda I: gamma*I

    for i in range(n):
        N = S[i] + I[i] + R[i]

        k1_S = dSdt(S[i], I[i])
        k1_I = dIdt(S[i], I[i])
        k1_R = dRdt(I[i])

        k2_S = dSdt(S[i] + k1_S*h/2, I[i] + k1_I*h/2)
        k2_I = dIdt(S[i] + k1_S*h/2, I[i] + k1_I*h/2)
        k2_R = dRdt(I[i] + k1_I*h/2)

        k3_S = dSdt(S[i] + k2_S*h/2, I[i] + k2_I*h/2)
        k3_I = dIdt(S[i] + k2_S*h/2, I[i] + k2_I*h/2)
        k3_R = dRdt(I[i] + k2_I*h/2)

        k4_S = dSdt(S[i] + k3_S*h, I[i] + k3_I*h)
        k4_I = dIdt(S[i] + k3_S*h, I[i] + k3_I*h)
        k4_R = dRdt(I[i] + k3_I*h)

        S[i+1] = S[i] + h/6*(k1_S + 2*k2_S + 2*k3_S + k4_S)
        I[i+1] = I[i] + h/6*(k1_I + 2*k2_I + 2*k3_I + k4_I)
        R[i+1] = R[i] + h/6*(k1_R + 2*k2_R + 2*k3_R + k4_R)

    return S, I, R

alpha0 = 58932628; alpha1 = 40380; alpha2 = 25610
h = 1; a = 0; n = 30; t = t_i(); beta =  0.4/58998618; gamma = 1/15
S, I, R = rk4_SIR(n)

data = {'Susceptible (S)':S,
    'Infectious (I)':I,'Recovered (R)':R, 'Population (N)': S + I + R}
SIR_table = pd.DataFrame(data).round()
```

Reference

Numerical Analysis 10th Ed. by R. Burden, J. D. Faires, A. M. Burden

"Novel Coronavirus (2019-NCoV) Situation Reports." *World Health Organization*, World Health Organization, 2020, www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports.