Tracking Covid-19 Infections in the US with the SIR Model in Python

Kenneth Huang
Department of Physics, University of California, Santa Barbara, CA 93106
(Dated: December 21, 2020)

The purpose of this modeling experiment was to simulate the SIR model using the runge kutta method, and fit the model prediction against real covid data containing daily new infections in the US. This process was implemented in python, and the line of best fit was generated with a built-in module called curve fitter. The fit initially determined optimal parameters values for the SIR model with uncertainties of $k = 0.1897 \pm 0.0029$, $\gamma = 0.018 \pm 0.004$, and $R_0 = 10.54 \pm 2.17$. While an R_0 greater than 1 is expected, this large of an R_0 likely indicates that the model is overestimating how quickly the virus spreads in the US. An extra parameter "c" was introduced to the SIR model to mitigate error in fitting the early portions of the data. In this modified model R_0 was extracted to be 41.30 ± 12.12, where $k = 0.2891 \pm 0.0193$ and $\gamma = 0.007 \pm 0.002$. As a result of this modification, error metrics significantly improved, while the uncertainties in R_0, γ, and k grew larger. This difficulty in extracting a reasonable R_0 value may be attributed to the volatility in γ, and its randomness during the first few weeks of data.

I. INTRODUCTION

A. Background

The Covid-19 virus is a respiratory disease that originated from China. Since March 2020, there has been extensive efforts from various countries to slow the spread of the virus, many of which have failed. As a result, various industries and businesses have been forced to shut down due to a widespread shelter-in-place order. The United States, for example, has been in a country wide quarantine for the last 6 months. During this time period, daily number of cases have oscillated but still continue to rapidly grow. Between September 29 and October 30, daily new cases grew by 132 percent [1]. Other countries, like Taiwan, were more successful in stifling the growth in daily new cases.

B. SIR Model

The SIR model is defined as a set of differential equations describing the relationships between variables S, I, and R. The variable represent susceptible, infected, and recovered or removed persons, respectively.

At the start of an infection, a population of size N will have a few people initially infected, zero people removed since no one has died or recovered yet, and nearly N people susceptible. At any point in time, S+I+R=N.

The change in susceptible population is determined by the amount of contact between existing susceptible and infected individuals. This can be described by: $\frac{d}{dt}S(t) = -k\frac{S(t)}{N}I(t)$

Where $S(t)/N$ is population density, k is a transmission rate or rate of infection, and $I(t)$ is infectious people.

The rate of infectious people can be described by: $\frac{d}{dt}I(t) = k\frac{S(t)}{N}I(t) - \gamma I(t)$

Here, gamma (γ) is a constant that determines how many people will recover from the infectious group.

Finally, the last differential equation for recovered people is formulated as: $\frac{d}{dt}R(t) = \gamma I(t)$

The ratio between k and y equate to a constant R_0 known as the basic reproductive number. This constant estimates the number of new infections per current infection. When R_0 is less than 1, the rate of I(t) is decreasing, usually representing that the population is on track to beating the pandemic. Conversely, when R_0 is greater than 1, the rate of infection is still increasing and has not reached an inflection point yet.

C. Runge Kutta Method

The aforementioned differential equations can be solved with the Runge Kutta Method. This method is used to estimate solutions for differential equations of the form: $\frac{dy}{dt} = f(y,t)$ where h is the time step h.

$$\mathrm{K}_1 = hf(x_n, y_n)$$
$$\mathrm{K}_2 = hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2})$$
$$\mathrm{K}_3 = hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2})$$
$$\mathrm{K}_4 = hf(x_n + h, y_n + k_3)$$
$$\mathrm{y}_{n+1} = y_n + k_1/6 + k_2/3 + k_3/3 + k_4/6 + O(h^5)$$

Runge Kutta Method

This method uses approximations to the slope to calculate a weighted sum for an estimate of $y(t)$ after n time steps [2]. Firstly, k_1 computes the inital slope, while k_2 calculates a more accurate slope using a second order midpoint method. The slope of k_2 is then used to move halfway through a timestep $(+\frac{h}{2})$ to provide another estimate of the slope, k_3. Lastly, k_3 is used to estimate the slope at the endpoint, resulting in the k_4 approximation. These k's are summed and weighted to find an approximation of the solution $y(t)$.

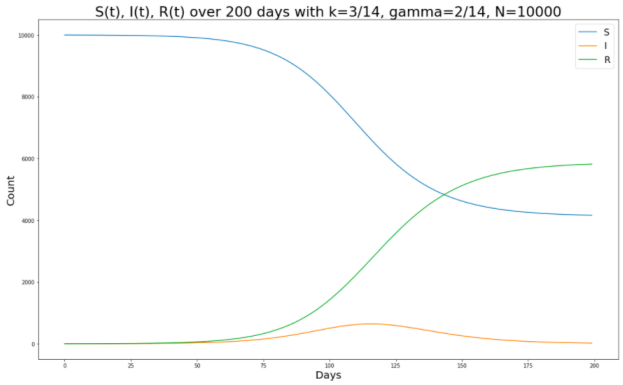

FIG1: The SIR differential equations solved with the runge kutta method

II. METHODS

A. Implementation

The runge kutta SIR model was programmed on a Jupyter notebook using python. The model contained parameters of t (time), k (transmission rate), gamma (recovery rate), and pop (population size). An additional parameter c (stifling constant) was later added to reduce error in the model. The model assumed $I_0 = 1$, $R_0 = 0$ (initially recovered or removed), and $S_0 = N - 3$ where N is total population of a particular country. Within the for loop that runs through 100 days, k_n's were calculated for S(t), I(t), and R(t). In other words, the runge kutta method was applied to the three of these differential equations to generate solutions.

For the purpose of this experiment, only solutions to I(t) were used for data fitting and analysis.

B. Fitting Techniques

Python's built in library, $scipy.curvefit$, was used to fit the model onto real data. This function takes in 5 parameters, then returns optimal model parameters and standard deviations of those parameters.

Parameters	Data Type
model function	f(x,...)
xdata	independent variable array
ydata	dependent variable array
initial parameters	array of guesses
initial bounds	tuple(s) for parameter bounds

This curve fitter module feeds the initial parameter values into the modeling function and runs through thousands of different values (within the provided parameter bounds). The module then finds a set of model parameters that minimizes the difference:

f(xdata, param1, param2, param3, ...) - ydata

In this particular data experiment, the module will find the best values for k, γ, and N that minimize the difference:

Model($xtimearray$, k, γ, N) - real data

This is essentially how python's curvefit module finds the best fitting line—by finding modeling parameters that generate the least amount of error with respect to the provided ydata.

C. Uncertainty in Parameters

The standard deviations in k, γ, N and all other parameters were calculated with the reduced chi square statistics, also known as a weight sum of squared deviations:

$$\chi^2 = \sum_{i=1}^{N} \frac{{r_i}^2}{{\sigma_i}^2}$$

Here, the residuals generated from each parameter is defined as r_i, where:

r_i = ydata - model($xdata$, $param_i$).

The σ is variance, which is set to be a N-length, 1-D array filled with 1's by default. This formula is used to compute standard deviations for k, γ, and all other SIR-model parameters.

D. Initial Parameter Selection

It is important to provide a decent estimate for the initial conditions; if unreasonable parameter values are provided, the curve fitter will compute standard deviations of 0 or infinity. To estimate "good" initial values for the model, several SIR solutions were plotted over 200 days using varying k's and γ's:

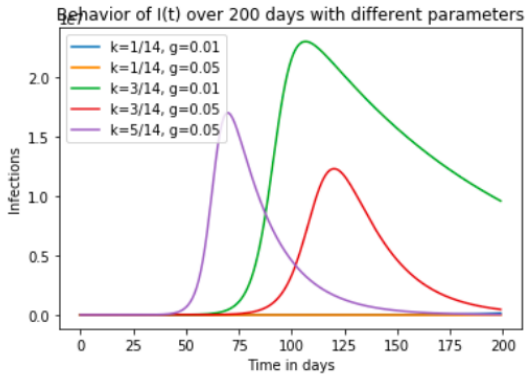

FIG2: SIR Model Behavior to estimate parameters

With US daily cases still on the rise, the curves of these SIR solutions did not accurately represent the growth of infections the US data. For time periods of 200 days or more, the SIR model assumes that the rate of infection is decreasing. Thus, only the first 100 days of data were be fitted in order to generate proper predictions.

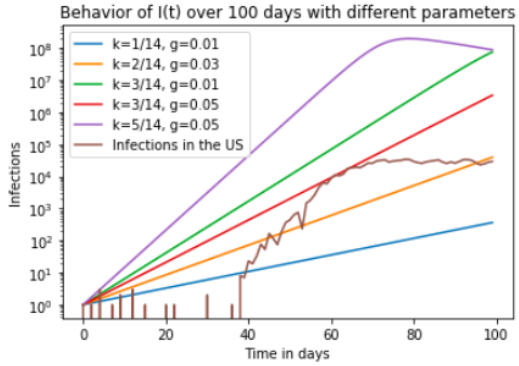

FIG3: A log plot was taken to analyze the proper domains for k and γ

From this 100 day log plot, the data to be fitted falls within solutions of the SIR model when 2/14<k<3/14 and 0.03<γ<0.05. These values will later be used for fitting.

Lastly, it may be helpful to specify initial bounds in the curve fitting module in case an unreasonable optimal parameter value is returned (i.e.: $\gamma = -0.5$).

E. Data Analysis

The first 100 days of new Covid-19 infections in the US were used to evaluate the SIR model. On January 22, 2020, the US saw its first covid infection and thus, $I_0 = 1$. With initial parameters of t = 0, k = 2/14, $\gamma = 0.01$, and population = 328,000,000, the curve fitter produced these results:

Parameter	Initial Guess	Optimized Parameter
k	0.143	0.1897 ± 0.0029
γ	0.01	0.018 ± 0.004
N	328,000,000	45820.0 ± 3262.0
R_0	14.30	10.54 ± 2.17

Feeding these optimized parameters in our SIR model and plotting the solution against the first 100 days of US data, resulted in a visually decent fit. Plotting the SIR model with $k = 0.1897 \pm 0.0029$ and $\gamma = 0.018 \pm 0.004$ produced a very narrow error spread.

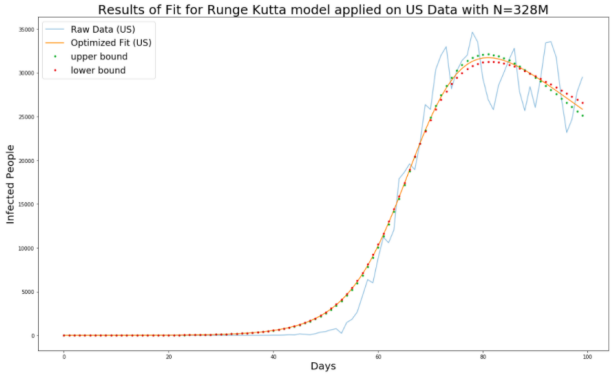

FIG4: Infections vs Days with Error Spread over 100 days. (Country: US)

While this solution somewhat captured the behavior of infection rates in the US, the fit error metrics was undesirably large. The mean squared error between the prediction and data was 4.33e6, and chi squared error was 4.38e4. These abnormally large errors are due to the first 40 days of erratic growth. This is apparent when plotting the same solution on a log scale.

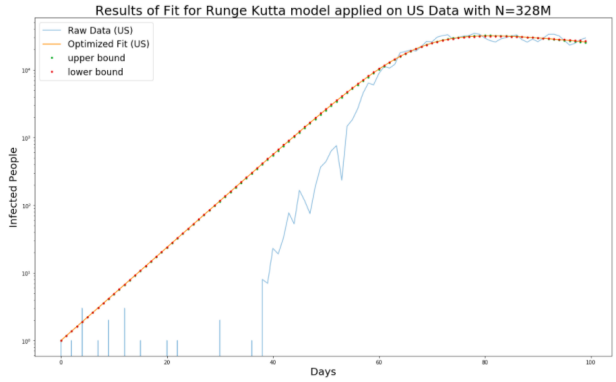

FIG5: Log Plot: Infections vs Days. Significant error is observed into the first 50 days. (Country: US)

F. Improved Model: Additional Parameter

From analyzing the log plot, the model grows too quickly during the first 40 days after the first infection and thus, overshoots the data and creates massive error.

To treat this, an additional parameter: c was introduced to the model. This parameter stifles the growth of infections for 40 days after the 1st initial infection.

In the model's for loop, each $I(t)$ term is multiplied by c = 0.995 for the first 40 days, then appended into the array containing solutions for I(t). From the 40th day to the 100th day, the parameter is neglected (c = 1). This factor of c should reduce the growth of the model in the first 40 days, while still accurately fitting the rest of the time period. Thus, the curve fitting module optimizes this extra parameter "c" for only the first 40 days of fitting.

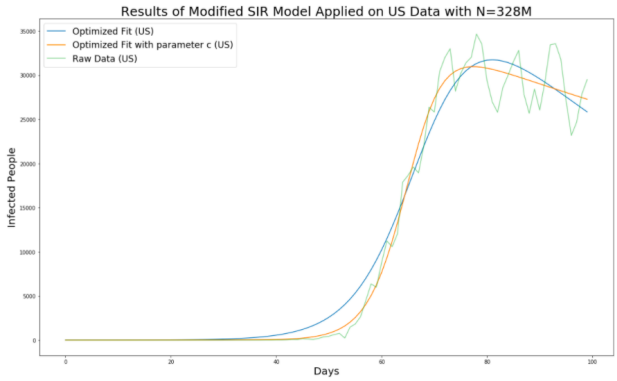

FIG6: I(t) over 100 days with optimal c = 0.866. (Country: US)

As seen from the plot above, the growth in the first 40 days is more accurate with the inclusion of c (in orange). This adjustment also significantly improved the error metrics, but worsened precision as seen from the error spread plot below.

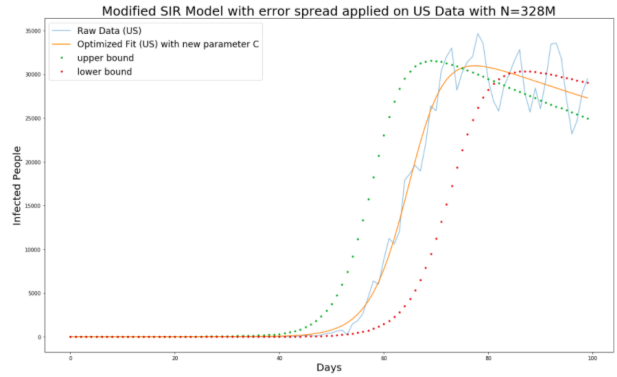

FIG6: Error spread of the I(t, k, γ, c, N) solution. (Country: US)

III. RESULTS

Introducing an additional parameter "c" resulted in the following SIR parameter values and error metrics:

Parameter	Initial Guess	Optimized Parameter
k	0.143	0.2891 ± 0.0193
γ	0.01	0.007 ± 0.002
c	0.995	0.8664 ± 0.0219
N	328,000,000	34680.0 ± 1420.0
R_0	14.30	41.30 ± 12.12

- mean squared error = 2.63e6
- chi squared error = 1.35e4

As a result of introducing the new parameter, mean square error was reduced by 39.26%, and chi squared error by 69.18%. However, the value and uncertainty of R_0 grew considerably. While it is expected for $R_0 > 1$ in the US, a R_0 of 41.30 is unreasonably large. For Sri Lanka, R_0 is estimated to fall between 0.93 and 1.23 [3], while China reported a R_0 value of 2.2 [5].

A possible of source of uncertainty and inaccuracy in R_0 may come from the gamma term (γ) in the SIR model. This constant determines the ability for an infected person(s) to recover and in the first 40-50 days of an infection, can be very difficult to predict. The gamma in the table above ($\gamma = 0.007 \pm 0.002$) might describe data where very few people are infected, like the case counts of Covid-19 in January or February. During this time period, very few people have been infected so there are very few people recovering, hence a very long recovery time.

IV. DISCUSSION

Factors like social distancing and government stringency may cause γ to fluctuate wildly throughout a pandemic. In the beginning phases of a virus spread, γ and k generate significant uncertainty.

FIG7: SIR model struggles to capture early behaviors of random, infectious growth in the first 50 days.

In this plot, it can be seen that the parameter "c" reduced the model's solution to fit closer to the actual data points. However, a more appropriate solution for the first 40 days would be the lower bound (in red), since the true solution still overshoots the raw data by quite a significant amplitude. Thus, the model is still quite inaccurate at predicting infections early on in a pandemic. This is largely due to the randomness of virus spread in the early stages; with 1 person initially infected, it takes a few weeks for "true" growth in infections to show in the data. In some cases, the infections may just dissipate before actually creating a pandemic.

Second, this plot indicates that a significant amount of uncertainty stems from fitting the first 40 days of data. The lower bound model solution estimates a later peak at around 80 days, while the upper bound solution indicates an earlier peak at around 60 days.

To extrapolate more accurate values for R_0 or γ, one may need to apply the SIR model to periodic segments in the data. The SIR model should use the "stifling constant" c, until the the infection data points start to noticeably grow. The latter portion of the data should omit this parameter, and run a normal SIR model with initial infections set to I_0 = 23 (Number of infections on the 40th day or day where obvious growth begins). This approach splits fitting the SIR model into two parts, which would return 2 different gammas, k's, and R_0's. Doing this should generate better error metrics and precision when fitting the two separate time periods.

V. CITATIONS

[1] https://ourworldindata.org/coronavirus/country/united-states

[2] https://lpsa.swarthmore.edu/NumInt/NumIntFourth.html

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7538841/CR

[4] https://github.com/CSSEGISandData/COVID-19

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7121484/