ODE Methods for Epidemiological Models

Ronan Perry
Biomedical Engineering
Johns Hopkins University
Baltimore, MD 21218
rperry27@jhu.edu

Christian Cosgrove
Computer Science Department
Johns Hopkins University
Baltimore, MD 21218
ccosgro2@jhu.edu

May 11, 2020

Abstract

Modeling the progression of disease in a population is a challenging task but critical to forecasting future dynamics, understanding potential causal mechanisms, and designing appropriate policy. It has been important historically, and is evermore pressing given the current COVID-19 epidemic. Simple models capture the basic dynamics, but accounting for longer-term dynamics and the effects of interventions is critical for any applications. We present a study of the SIR model, its extension to the SIRS model, and the addition of intervention dynamics as applied to New York City COVID-19 data.

1 Compartmental Models: Background

In this project, we study a class of epidemiological models called *compartment models*. While powerful, this class of models relies on the following key assumptions:

- **The population is large**. Rather than treating the population as a collection of discrete individuals, we treat the number of infected individuals as a continuous real value.
- **The dynamics of infection are deterministic**. While the underlying dynamics of disease spread may be random, we treat the overall spread of the disease as deterministic. This assumption can be justified by the previous (large-population) assumption, e.g., using the law of large numbers.[1]
- **At each time point, every individual is in one state**. In epidemiological models, a state could indicate that the individual is infected and can spread the disease, or that they are immune, among others.

1.1 An example

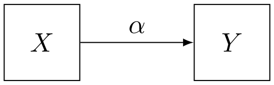


Figure 1: A toy example of a compartment model with two states, X and Y.

The model shown in Figure 1 corresponds to a system with two states, where individuals in state X move to state Y at a rate α. Because the rate itself is also proportional to the number of individuals in the state, this corresponds to the ODE:

$$\frac{dX}{dt} = \alpha X$$
$$\frac{dY}{dt} = -\alpha X$$

[1] This type of reasoning is common in statistical physics.

Note that the total number of individuals, $N = X + Y$, is conserved:

$$\frac{dN}{dt} = \frac{dX}{dt} + \frac{dY}{dt} = \alpha X - \alpha X = 0$$

This balancing property is necessary for the compartmental model to be well-defined. Thus, even though there are two states, the effective number of degrees of freedom is one.

Note: The rate α can itself depend on other variables, as we will see in the next section.

2 Epidemiological Models

We begin by introducing the following key parameters [1].

- β: Number of infections per infected individual per unit time (units: 1/time)
- γ: Rate of recovery (units: 1/time)

An important term derived from these is R_0, the total number of secondary infections caused by an infectious individual in a susceptible population during the course of their infectious period. In the simplest models, we view it as a fixed constant and it indicates the stability of the system equilibrium.

$$R_0 = \frac{\beta}{\gamma}$$

2.1 SIR model

The *Susceptible-Infectious-Recovered* (SIR) model is a three-compartment model. There are three states [1]:

- S: Susceptible. An individual can be infected.
- I: Infectious. An individual can transmit the infection.
- R: Recovered. An individual is immune to infection.

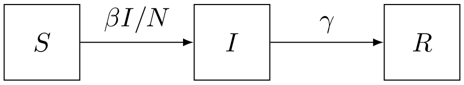
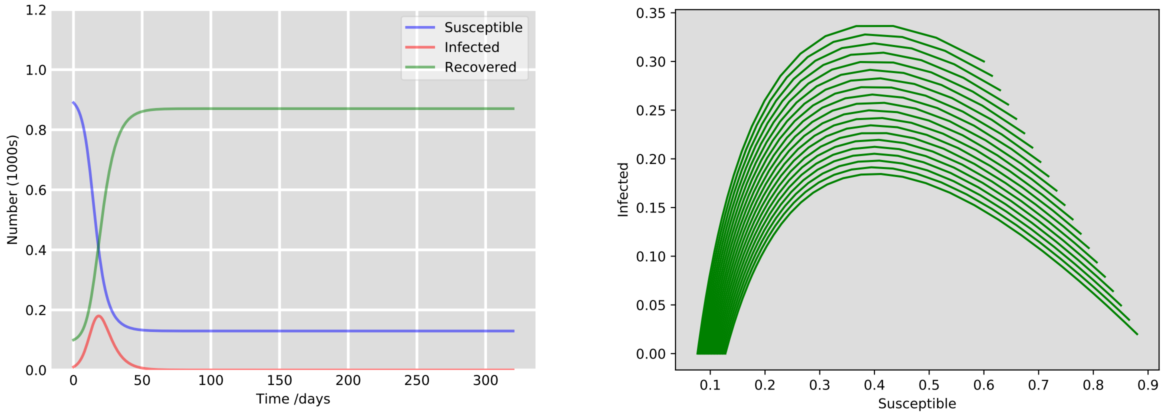


Figure 2: SIR model



(a) Temporal dynamics

(b) Phase portrait

Figure 3: Simulated dynamics with the SIR model. Notice that the number of infections initially grows exponentially, but eventually tends towards an equilibrium value of zero.

2.2 Decaying immunity (SIRS)

The SIR model assumes that after an individual recovers, they are immune forever. While this is applicable to certain diseases (e.g., measles), it is not applicable to diseases with temporary immunity (i.e. the seasonal flu).

We can model the decay in immunity by adding an additional process that transforms recovered individuals to susceptible at a rate ζ: [1]

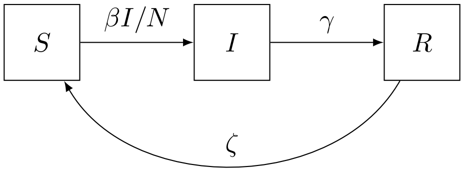
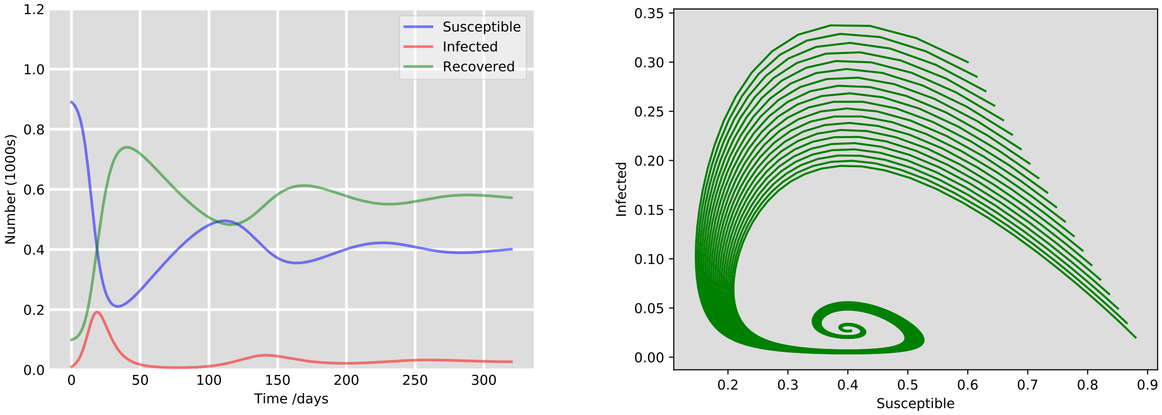


Figure 4: SIRS model

This corresponds to the final ODE

$$\frac{dS}{dt} = -\frac{\beta IS}{N} + \zeta R$$
$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I$$
$$\frac{dR}{dt} = \gamma I - \zeta R.$$

Note, if $\zeta = 0$ we recover the SIR model and system of ODEs. This is a two-dimensional model and like the SIR model cannot be solved analytically. Phase portraits and simulations must be used to estimate the solution.



(a) Temporal dynamics

(b) Phase portrait

Figure 5: Simulated dynamics with the SIRS model. Note that temporary immunity leads to cycles of infection, which are characteristic of the flu and other seasonal outbreaks.

2.3 Further extensions

Besides birth and deaths, time-dependent parameters (i.e. due to seasonality), additional compartments (i.e. vaccinated), and the inclusion of heterogeneous groups can all be added to these models. An extension which we will consider is time-varying infection rates in the SIR model. Specifically, consider the modified ODE

$$\frac{dI}{dt} = \frac{r\beta I}{N}\pi(t)$$

where $\pi(t)$ is a scalar in $[0, 1]$ modulating the rate of infection.

3 Numeric Approximations

Suppose we have a generic ODE

$$\frac{dy}{dt} = f(t, y)$$

with initial condition $t = t_0$ and $y = y_0$. For many such ODE initial value problems, there is no well defined analytical solution. In such cases, one may instead rely on numerical approaches to generate solution trajectories from fixed starting conditions. One must be wary of such approaches, however, as oftentimes such trajectories are not stable to perturbations of initial conditions and diverge quickly from the true trajectory if not calculated carefully or estimated too far into the future.

One approach to numerically solving such ODEs is to approximate them at discrete time intervals. We let $y_i \approx y(t_i)$ for integer i where $t_i = t_{i-1} + h$ and h is a small time step. Such solutions can be approximated through linear multistep methods, of the form

$$\sum_{j=0}^{s} a_j y_{n+j} = h \sum_{j=0}^{s} b_j f(t_{n+j}, y_{n+j}),$$

with coefficients a_j and b_j.

3.1 Euler's method

The simplest of such multistep solutions is the forward Euler's method, a naive time-stepping solution of the form

$$y_{n+1} = y_n + hf(t_n, y_n).$$

At each point, the known slope is used to approximate the next point. Although simple to implement and understand, for more complicated systems it can fail to account for changes at higher derivatives.

3.2 Adams-Bashforth method

One problem with Euler's method is that it fails to account for how the $f(t, y)$ changes from t_n to t_{n+1}. The Adams-Bashforth (AB) method [2] approaches this problem by interpolating f over k points using a Lagrange polynomial

$$p(t) = \sum_{j=1}^{k} \frac{\prod_{t_i \neq t_{n-j}} (t - t_{n-j})}{h^{k-1}} f(t_{n-j}, y_{n-j})$$

in time. The next next step is

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y) dt$$

and so we can substitute our polynomial $p(t)$ for $f(t, y$ which yields an approximate solution for y_{n+1}.

Note that the AB method can either integrate to t_n, what is termed an explicit method as y_n is known, or to t_{n+1}, what is termed an implicit method [3]. In the latter case, it is actually the Adams-Moulton (AM) Method [2]. This choice leads to two separate variants of the algorithm with slightly different properties, notably in terms of stability, but the same overall concept.

3.3 Backwards Differentiation Formula (BDF) method

BDF [2] is akin to the AM method but takes a different approach, using a kth order interpolating polynomial over the y_j points and taking the derivative. The Lagrange polynomial over the interval $[t_{n_1-k}, t_{n+1}]$ is

$$p(t) = \sum_{j=0}^{k} \alpha_j(t) y_{n+1-j},$$

where $\alpha(t)$ are the time-dependent Lagrange coefficients. We can then take the derivative and set it equal to f at t_{n+1} per our given ODE equation, resulting in

$$\frac{d}{dt} p(t) \mid_{t_{n+1}} = \sum_{j=0}^{k} \frac{d}{dt} \alpha_j(t) y_{n+1-j} \mid_{t_{n+1}} = f(t_{n+1}, y_{n+1}).$$

At this point, we can solve for y_{n+1}.

One of the benefits of the BDF Method is that with the same order as an Adams Method, it achieve a degree higher error term, leading to faster decay with smaller step size [4]. It thus allows us to consider larger time steps at the same order with the same accuracy, a computational benefit. A cost of BDF, however, is that by being an implicit method it requires solving a potentially complicated system to identify y_{n+1} [4]. The more complicated the function f is, the more difficult the implicit solution becomes.

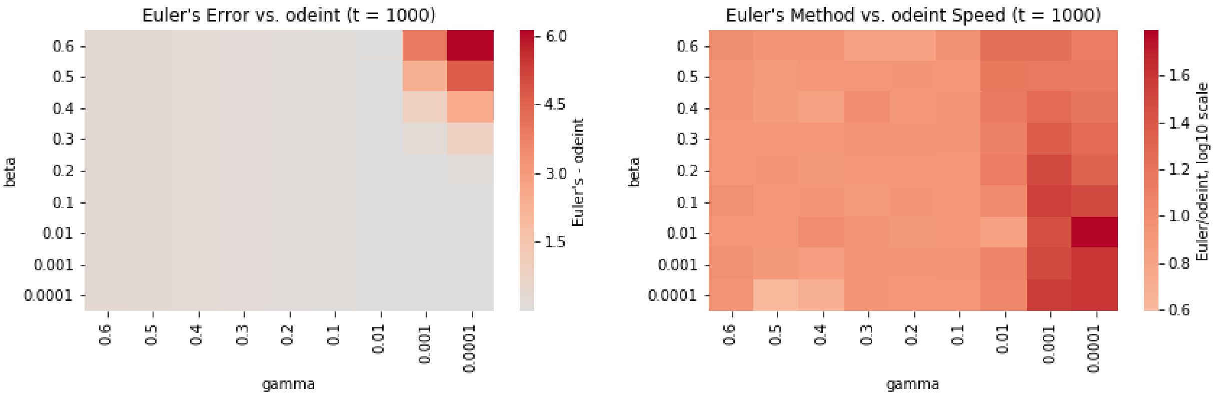
3.4 Stiffness

As mentioned before, complicated ODEs may pose issues of stability in approximating their solutions with numerical methods. Such difficult ODEs to approximate are termed "stiff". In our cases, a stiff ODE refers to one for which all the eigenvalues have negative real components and for which the ratio between the norm of the largest eigenvalue and the smallest is large [4] [2]. Effectively, small perturbations in initial conditions may result in large variations from the true trajectory. Whether the ODE in hand is stiff or non-stiff is a reason to pick an implicit or explicit method.

If the problem is not stiff, a solver such as the explicit Adams Method is easier to solve while not sacrificing stability of its solution. Under stiff problems, however, the explicit method is no longer potentially stable and so BDF may be employed as an implicit solution. The computational burden of a potentially complicated function f is the penalty for a more stable solution.

4 Simulations

We ran our analyses and simulations in Python using the SciPy [5] library to compute numerical approximations to our systems. Due to our need for complicated models with time-varying parameters and since it was easier to implement our own time-stepping method to solve those, we compared the accuracy and speed of our time-stepping (Euler's Method) SIR model to one solved using the SciPy *odeint* function. The *odeint* function utilizes the LSODA method, a wrapper of a solver in the Fortran ODEPACK [6]. LSODA is an adaptive combination of the Adams-Moulton and BDF Methods, switching between the two if the system seems to become stiff.



(a)

(b)

Figure 6: Difference in performance (a) and speed (b) between the Euler's ethod and *odeint* solutions. The Euler's method solutions seems to only lose performance when gamma is small but beta is big. The Euler's method seems to be slower at all points than *odeint*.

The results in Figure 6 reveal the change in performance using our Euler's method versus the *odeint* function. The Euler's method solutions seems to only lose accuracy when gamma is small but beta is big, presumably because this leads to rates at different time scales. The Euler's method seems to be slower at all points than *odeint*, presumably because our implementation is not efficient and the problem is simple enough that *odeint* doesn't induce any computational hurdles.

5 Applications to COVID-19

One of the most clear applications of these epidemiological models is examining the current COVID-19 pandemic. The SIR model offers a good fit [7] [8] and at this time there is not a clear need for either an exposed compartment or fluctuating immunity. However, the raw dynamics of COVID-19 have been disrupted largely in part by social-distancing interventions, leading to decrease in human contact and thus rates of infection.

Our goal is to model the dynamics of the pandemic under an SIR model with exponentially decaying contact

$$\pi(t) = (1-\alpha)e^{-ct} + \alpha$$

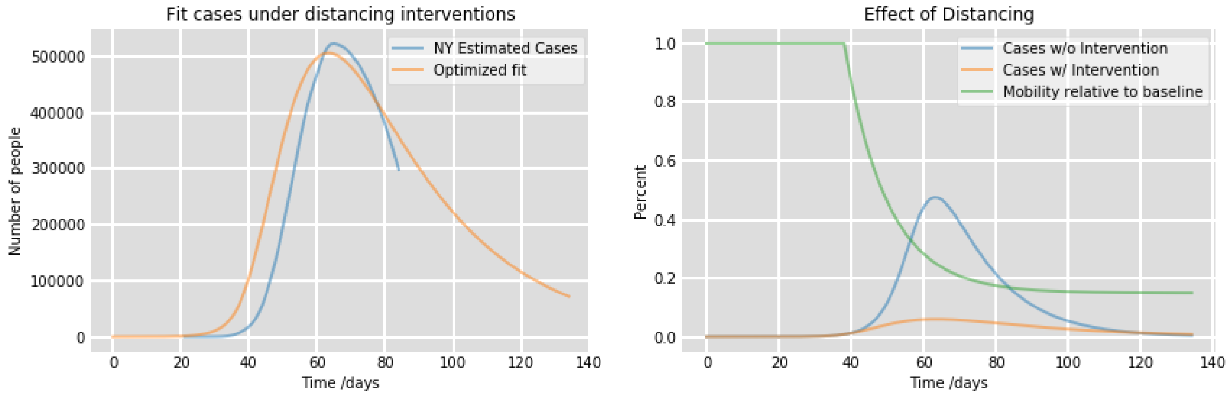
which is 1 at $t = 0$ and asymptotes to α with rate determined by c. We incorporated this into our forward Euler's method implementation as our simulation results indicated sufficient accuracy given the magnitudes of our parameters.

We acquired our data from the publicly accessible New York City (NYC) Health webpage [9] which provides estimated counts of the number of cases, hospitalizations, and deaths due to COVID-19 from March 3rd onward out of the approximate total population of 8.96 million. However, it is increasingly clear that such counts underestimate the true effects, especially the number of true cases as the US has lagged heavily in its testing capabilities. While deaths still undercount, they can potentially provide a better estimate of the progression of cases. While the true death rate is not known, a lower bound has been estimated to be 0.5% with upper bounds around 2% for certain at-risk populations [10]. We use an approximate death rate of 1% and scale the available number of NYC deaths to approximate the progression of cases. We smoothed the data using a 3 day moving average, and then obtained the cumulative daily totals by summing the processed daily new cases with a fixed γ of $1/14$, meaning that a new case exited the infected compartment after 14 days [7].

We fit the model using a SciPy optimizer with modified, asymmetric least squares loss function

$$L(x, \hat{x}) = \begin{cases} (x-\hat{x})^2 & x \leq \hat{x} \\ 20(x-\hat{x})^2 & x > \hat{x} \end{cases}$$

to encourage overshooting the true data x rather than undershooting, as underreporting of cases is much more likely.



(a)

(b)

Figure 7: (a) Fitted SIR and decaying contact model to the NYC data predicts an R_0 of 4.68. (b) The fitted change in contact curve is shown, as well as the trajectory of cases without it per the fitted R_0.

With this model, we obtained a well-approximating SIR model and contact rate function. In Figure 7a, the fitted model is displayed along with the estimated, smoothed NYC case count. In Figure 7b, the decaying contact factor is shown, accounting for change in behavior due to social and legal interventions, as well as the hypothetical trajectory were such interventions not in place. This fitted model calculates an R_0 of 4.68. This is certainly higher than some estimates [11] but within other estimates [8]. The error is most likely due to the high density of New York City as well as underreporting, especially closer to the start of the outbreak.

Our estimated R_0 of 4.68 is plotted in Figure 8 alongside the estimated R_0 per timestep from the NYC Data using the 1 step difference equation

$$I_{t+1} - I_t = \frac{\beta_t S_t I_t}{N} - \gamma_t I_t$$

with fixed $\gamma = 1/14$ as before.

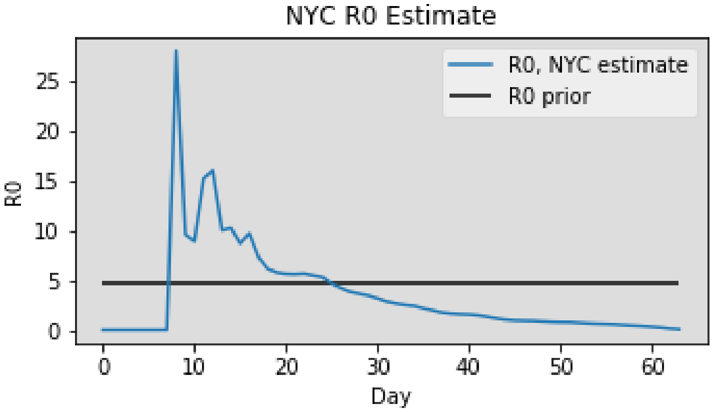


Figure 8: Daily estimated R_0 from the adjusted NYC data. Estimated $R_0 = 4.68$ plotted for reference. The wild values and uncertainty at the beginning may reflect underreporting of cases and a poor understanding of when the epidemic actually began in the city.

6 Discussion

We have presented two extensions to the SIRS model with applications to data of the COVID-19 outbreak in NYC. Both of these models better capture real-life dynamics and enable the construction of models with a stronger capability to fit data we observe. Our intervention-adjusted model appears to capture the NYC dynamics moderately well. However, this required numerous assumptions and it is clear that the data itself is extremely noisy. Construction of more accurate and trustworthy models will weigh heavily on our ability to capture cleaner data and make sounder assumptions.

References

[1] Institute for Disease Modeling (ICM). Sir and sirs models. EMOD Documentation.

[2] Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Stat. Comput.*, 4(1):136–148, March 1983.

[3] Varun Shankar. Linear Multistep Methods I: Adams and BDF Methods. page 11.

[4] Varun Shankar. Linear Multistep Methods II: Consistency, Stability and Convergence. page 6.

[5] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[6] A. C. Hindmarsh. Odepack, a systematized collection of ode solvers. *IMACS Transactions on Scientific Computation*, 1:55–64, 1983.

[7] Peter X Song, Lili Wang, Yiwang Zhou, Jie He, Bin Zhu, Fei Wang, Lu Tang, and Marisa Eisenberg. An epidemiological forecast model and software assessing interventions on COVID-19 epidemic in China. preprint, Infectious Diseases (except HIV/AIDS), March 2020.

[8] Yi-Cheng Chen, Ping-En Lu, Cheng-Shang Chang, and Tzu-Hsuan Liu. A Time-dependent SIR model for COVID-19 with Undetectable Infected Persons. *arXiv:2003.00122 [cs, q-bio, stat]*, April 2020. arXiv: 2003.00122.

[9] NYC Health. Covid-19: Data. 05/07/2014.

[10] Chirag Modi, Vanessa Boehm, Simone Ferraro, George Stein, and Uros Seljak. Total covid-19 mortality in italy: Excess mortality and age dependence through time-series analysis. *medRxiv*, 2020.

[11] Sheng Zhang, MengYuan Diao, Wenbo Yu, Lei Pei, Zhaofen Lin, and Dechang Chen. Estimation of the reproductive number of novel coronavirus (COVID-19) and the probable outbreak size on the Diamond Princess cruise ship: A data-driven analysis. *Int. J. Infect. Dis.*, 93:201–204, April 2020.