QBioS Laboratory on Outbreaks: From Prediction to Control

Joshua S. Weitz[1, 2, *]

[1] School of Biology, Georgia Institute of Technology, Atlanta, GA, USA
[2] School of Physics, Georgia Institute of Technology, Atlanta, GA, USA

(Dated: May 13, 2021)

I. OUTBREAKS: FROM DETERMINISTIC MODELS TO STOCHASTIC REALIZATIONS

How do public health scientists use models of outbreaks to forecast the shape of outbreaks to come, to interpret the shape of outbreaks that have already taken place, and/or to determine how best to intervene? This laboratory provides a direct path towards building dynamic simulation approaches to support the control and prevention of infectious disease. In doing so, the laboratory will leverage the core ideas of this morning's lecture to transform epidemic theory into simulation dynamics. These laboratory notes span a five year development process, from the Fall of 2015 to the Fall of 2021. In doing so, they span the Ebola virus disease outbreak in West Africa to the global Covid-19 pandemic. In Fall 2015, it was already apparent that computational models could be used in real-time to support outbreak responses. That is: relatively simple models were used to make projections, guide interventions, and estimate parameters that could connect real-world data and ongoing issues of control. The issues of how to connect population models is ongoing; there have been multiple EVD outbreaks since then, including one in 2018 in the Democratic Republic of Congo, with some concern that these epidemics will become endemic. Nonetheless, there is also optimism, given the development of the 'rVSV-ZEBOV' vaccine that protects against the Zaire strain (the same strain that was the causative agent of the 2014-2015 outbreak). In contrast, theory and simulation along with model-data integration have taken on an every larger role in shaping the initial and sustained response to a global Covid-19 pandemic. Models have been critical in shaping responses, but the often simplified choices made have also been criticized, given the need to take actions based on inference from intentionally simplified or incomplete representations of the real world.

Indeed, it is critical to keep in mind that variants of the kinds of models developed in this laboratory are used, in practice, by academic modeling groups, governmental agencies (like the WHO and CDC), and independent non-governmental organizations to help inform policy makers as they decide how best to allocate resources, design vaccination programs, public health campaigns, and other interventions to prevent and control disease. Indeed, one could argue that despite their simplicity, this laboratory goes above and beyond some intervention-level tools that utilize curve fitting approaches. For example, widely influential early Covid-19 models in Spring/Summer 2020 developed by the Institute for Health Metrics and Evaluation used curve fitting approaches as a means to predict the waning of the pandemic from case data. These early estimates were overly optimistic and ultimately flawed, largely because they did not include mechanistic representations of the underlying disease states of the population. In contrast, the kinds of models developed here have greater flexibility but do require an expanded suite of methodological tools. The core equations here include feedback between different types of individuals, whether susceptible, infectious, recovered, or removed, as well as potential interventions to move individuals from one state to another (e.g., via isolation) or to change interaction rates between individuals (e.g., via masking or social distancing). By using a high-level programming language this laboratory develops a suite of flexible approaches including dynamic and stochastic models. These models are also more robust – technically. The state of modeling in response to the Covid-19 pandemic reflects a maturation of real-time response efforts: including epidemic models, model-data integration, scenario building, real-time estimators, and more. This lab can't do all of that. But, it will provide the core principles to help you move beyond armchair epidemiology status. The core methods here form the basis for state of the art response models found in the rapid proliferation and interest in public health globally.

In brief, today's lab will teach you how to (i) model an susceptible-infectious-recovered ('SIR') epidemic outbreak; (ii) explore the relationship between speed, strength, and size of an outbreak; (iii) develop a stochastic simulation of disease dynamics, flexible enough to be applied to real-world cases. This is a lot - but it builds on concepts established in the morning lecture and that form the basis for real-world models.

A pre-final note about this format. The laboratory here represents an adaptation of a laboratory used in the 'Foundations of Quantitative Biosciences' class, taught every Fall at the Georgia Institute of Technology. This class is

*Electronic address: jsweitz@gatech.edu; URL: http://ecotheory.biology.gatech.edu

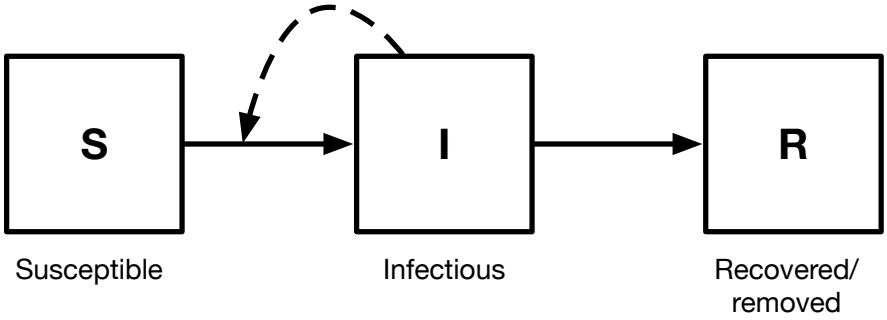


FIG. 1: Disease dynamics represented in terms of susceptible, infectious, and recovered/removed individuals, i.e., a SIR model. Infectious individuals can come into contact with susceptible individuals leading to new infections. Infectious individuals recover or are removed from circulation (in the case of potentially fatal diseases). These interactions form the basis for the rates and dynamics in both deterministic and stochastic versions of population-level epidemic models.

the cornerstone class for the PhD in Quantitative Biosciences (or QBioS). The laboratory and accompanying textbook are in the final stages of development (hint: watch the QBioS website for more info - `https://qbios.gatech.edu`). As we remind students in the QBioS cohort each Fall - these laboratories are meant to bridge the gap between concepts and application. The answers are not here at the outset precisely to give you a chance to explore, make (purposeful) mistakes, and to find your own path to understanding. And yes, we will give you the answers (aka the 'instructor version') at the end of the day as a resource for refining and deepening understanding.

A final note: you can do it!

II. EPIDEMIC MODEL - FUNDAMENTALS

A. Transitions to Outbreaks

Consider an infectious diseases that can spread in a population of size N. Individuals may be susceptible, infectious, or recovered/removed. We term these individuals S, I and R respectively, such that $S + I + R = N$. The re-scaled dynamics can be written as

$$\begin{aligned} \dot{S} &= -\beta SI \\ \dot{I} &= \beta SI - \gamma I \\ \dot{R} &= \gamma I \end{aligned}$$

given the two processes, infection at a rate βSI and recovery at a rate γI. The main text includes the full derivation of these equations and the meaning of β. Note also that variability in susceptibility can lead to fundamental changes in these equations. Here, this model denotes the fraction of individuals of each type, i.e., $S + I + R = 1$. Simulations of this model should use the following core code, beginning with the representation of the ODE:

```
def sir_model(y,t,pars):
    """
    function dydt = sir_model(y,t,pars)
    SIR Model
    """

    S = y[0]
    I = y[1]

    #The model
    dSdt = -pars['beta']*S*I
    dIdt = pars['beta']*S*I-pars['gamma']*I
    dRdt = pars['gamma']*I
```

```
    dydt = [dSdt, dIdt, dRdt]
    return dydt
```

Applying this model requires a relevant choice of epidemiological parameters. Consider susceptible individuals that interact with c individuals per unit time, of which I/N are infectious, and of which a fraction p of such contacts lead to a new infectious event. In that case we expect the rate of transmission per susceptible individual to be cpI/N multiplied by the number of susceptible individuals to yield $cpSI/N$ or $\beta \equiv cp$.

Challenge Problem: Outbreak Criteria

Complete the simulation code below for the spread of an infectious disease beginning with 1 individual out of 10000 and estimate the value of $\mathcal{R}_0$. Do you expect the disease to spread or not? Then, change the value of p to 0.01 – will the disease spread, why or why not?

```
# main data goes here
from scipy import integrate
pars={}
pars['c'] = 20 #Contacts per unit time (days)
pars['p'] = 0.025 # Probability of infectious contact
pars['beta']=... #transmission rate
pars['gamma']=1.0/4 #Recovery rate (1/days)
pars['basR0'] = ...
pars['N'] = 10000
pars['I0'] = 1
pars['S0'] = pars['N']-pars['I0']

# Run the model
t=np.arange(100)
y = integrate.odeint(sir_model,
       np.array([pars['S0'],pars['I0'], 0])/pars['N'],t,args=(pars,))

#Plot the results
plt.plot(t,y)
plt.xlabel('Time (days)')
plt.ylabel('Population fraction')
```


B. Speed, Strength, and Size

Once an outbreak starts, how far will it go? That is to say: how many individuals in a population will become sick? The relationship between speed, strength, and size is critical to understand outbreak dynamics. Let's define each term. The speed can be measured in terms of the rate, r, of exponential growth of the outbreak, i.e., $I(t) \sim e^{rt}$. The strength can be measured in terms of the dimensionless number $\mathcal{R}_0$. This value is known as the basic reproduction number. It denotes the average number of new infections caused by a single infectious individual in an otherwise susceptible population. Finally, the size of the outbreak can be measured in terms of the fraction of the population infected at the culmination of the outbreak, $R(t \to \infty)$, which is equivalent to $1 - S(t \to \infty)$. The main text presents a derivation of the strength-size relationship, specifically:

$$\mathcal{R}_0(S_\infty - 1) = \log(S_\infty) \tag{1}$$

as well as the strength-speed relationship, i.e.,

$$r = \gamma(\mathcal{R}_0 - 1) \tag{2}$$

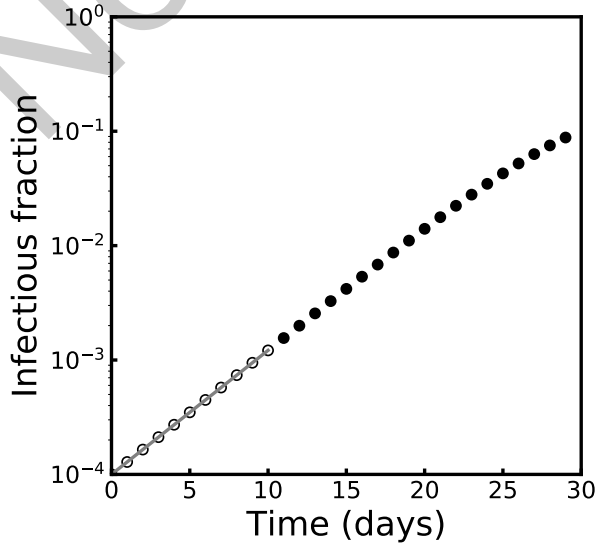
Here, you will explore both of these systematically, gaining an intuition for these two relationships.

1. Strength and Speed

It would seem that outbreaks with higher values of $\mathcal{R}_0$ would lead to a faster increase in case counts. But that is not necessarily the case. One way to explore this is to modulate both β and γ, i.e., the transmission and recovery rates, which influence both strength and speed. To begin, let's first measure the speed in an outbreak. The code below simulates an outbreak using $\mathcal{R}_0 = 2$, $\beta = 0.5$ days^{-1}, and $\gamma = 0.25$ days^{-1}. For this combination, we expect $r = 0.25$ days^{-1}. The following code illustrates the following steps. First, simulate the model. Second, fit a line to the log-transformed infectious counts, $I(t)$. Note that you could also find the line of best fit to the cumulative case incidence, $I(t) + R(t)$. The slope of this line should be the estimated $\hat{r}$.

```
#Run the model over 10 days
t=np.arange(11)
y = integrate.odeint(sir_model,
                     np.array([pars['S0'], pars['I0'], 0])/pars['N'],t,args=(pars,))

# Find the slope
p = np.polyfit(t,np.log(y[:,1]),1)

# Plot the data and overlay the best-fit exponential
fig = plt.figure()
ax = fig.gca()
plt.scatter(t,y[:,1],color='k',facecolor = 'none')
ax.set_yscale('log')
plt.plot(t,np.exp(p[0]*t+p[1]),color='r',linewidth=2)

# Use solid points for the future k
t = np.arange(30)
y = integrate.odeint(sir_model,
                    np.array([pars['S0'],pars['I0'],0])/pars['N'],
                    t,
                    args = (pars,))
tmpi = np.argwhere(t>10)
plt.scatter(t[tmpi],y[tmpi,1],color='k')
```

When implemented, this code results in the following image:



As is apparent the fraction of infectious individuals does grow exponentially to begin, at precisely the rate expected in theory. In this case, the slope of the line is 0.2496 days^{-1} and the expected value is $r = 0.25$ days^{-1}. With this code snippet in mind, your next challenge is to examine the speed-size relationship given variation in β and γ.

Challenge Problem: Strength-Speed Relationships

Determine the strength and speed for the following sets of disease parameters

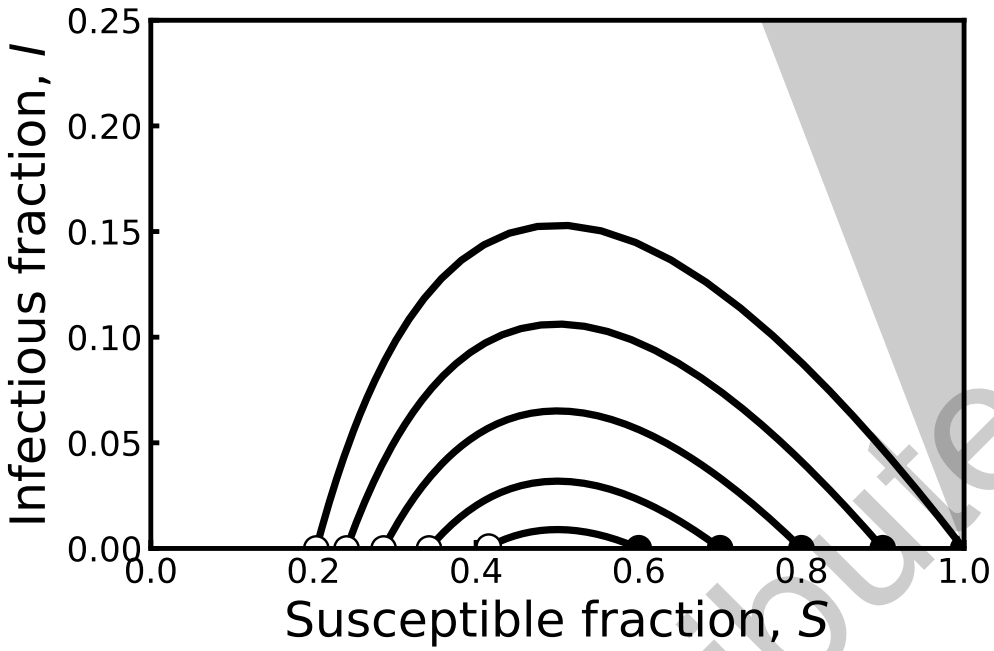
Transmission β	Recovery γ	Strength $\mathcal{R}_0$	Speed r
0.5	0.4	...	...
1	0.5	...	...
0.25	0.5	...	...
0.75	0.25	...	...

2. Strength and Size

An outbreak is not expected to infect all individuals. The final size of the epidemic in the idealized, homogeneous case can be measured in terms of the cumulative case count, i.e., the value of $R_\infty = 1 - S_\infty$. The next code set shows how to systematically modify the initial fraction of infected. In these overlaid simulations, the initial values are marked by black circles (on the right side of each arc) and the final points by gray circles (on the left of each arc). The dynamics in the phase suggest there is a critical value of $1/\mathcal{R}_0$ where an initial outbreak will not spread. This observation is the basis for findings that not all individuals in population must be vaccinated for a vaccine to be effective at the population level! However, an ancillary finding is that the fraction of the population that should be vaccinated increases with the strength of the disease, i.e., measured in terms of $\mathcal{R}_0$.

```
#Modify the initial values
pars['N'] = 10000
pars['S0_range'] = np.array([0.6, 0.7, 0.8, 0.9, 0.999])
pars['I0_range'] = 1/pars['N']*np.ones(5)
pars['R0_range'] = 1 - pars['S0_range'] - pars['I0_range']

# Run the model
fig = plt.figure()
ax = fig.gca()
plt.xlim([0,1])
plt.ylim([0,0.25])
for i in range(len(pars['S0_range'])):
    t = np.arange(200)
    y = integrate.odeint(sir_model,
                         [pars['S0_range'][i], pars['I0_range'][i], pars['R0_range'][i]],
                         t,
                         args=(pars,))
    plt.plot(y[:,0],y[:,1],color='k',linewidth=3)
    plt.scatter(y[-1,0],y[-1,1],color='r',s=100)
    plt.scatter(y[0,0],y[0,1],color='k',s=100)

#Show the excluded regime
from matplotlib.patches import Polygon
verts = [(1,0), (1,0.25), (0.75,0.25)]
poly = Polygon(verts, facecolor=[0.8,0.8,0.8])
ax.add_patch(poly)
```

Running this code yields the following trajectories where the shaded region denotes inaccessible parts of phase space given that $S + I \leq 1$.



This example of visualizing outbreak dynamics in the phase plane also provides the basis for the next challenge problem.

Challenge problem: Strength-Size Relationships in Phase Space

First, fix the value of the recovery rate to $\gamma = 0.25$, then modulate the transmission rate from $\beta = 0.25$ to $\beta = 2.5$. Assume that initially, 80% of the population is susceptible and 20% is recovered. In this case, does the disease always spread? Why or why not? In addition, find the outbreak size and show how it relates to $\mathcal{R}_{eff}$ – the effective reproduction number given the initial susceptible population.

C. From Outbreaks to Endemics

The SIR model, with rapid loss of immunity can be written in terms of an SI model:

$$\dot{S} = -\beta SI + \gamma I$$
$$\dot{I} = \beta SI - \gamma I$$

This model assumes that there is no immunity. Instead, recovered individuals are immediately susceptible. As a result, the dynamics include a new long-term possibility: an endemic disease state where $S^* = \frac{1}{\mathcal{R}_0}$ and $I^* = \frac{\mathcal{R}_0 - 1}{\mathcal{R}_0}$. This SI model can be coded as follows:

```
def si_model(y,t,pars):
    """
    function dydt = si_model(y,t,pars)
    SI Model
    """

    S = y[0]
    I = y[1]

    # The model
    dSdt = -pars['beta']*S*I + pars['gamma']*I
    dIdt = pars['beta']*S*I - pars['gamma']*I

    dydt = [dSdt, dIdt]
    return dydt
```

As should be apparent, there is only an endemic state with $I^* > 0$ when $\mathcal{R}_0 > 1$.

Challenge problem: From Epidemic Outbreaks to Endemics

Using $\beta = 0.3$ and $\gamma = 0.25$ days^{-1} show the convergence of the dynamics to an endemic state, i.e., from an outbreak to persistent infectious cases. And then compare and contrast this with dynamics in which immunity is permanent. For these parameters, simulate the dynamics over a 1 year = 365 day period.

III. STOCHASTIC EPIDEMICS

A. A Pre-Amble on Stochastic Simulations

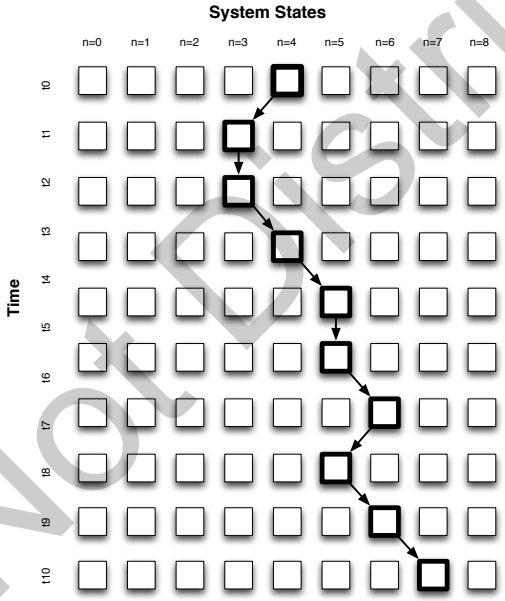
The stochastic simulation of epidemics will consider how a discrete set of individuals change amongst susceptible, infectious and recovered. To do so, we need to keep track of the state of each individual, and not the average, i.e., in an individual-based model there are 0, 1, 2, etc. individuals who are sick and not 1.2342341593. Simulating such individual-level dynamics requires an approach in which the state of each individual changes discretely, albeit in continuous time. To do so, we will use a method commonly known as the Gillespie algorithm.

The Gillespie algorithm is a widely used approach developed for simulating chemical reaction kinetics, one event at a time. It can be used in many contexts, including stochastic gene expression, population dynamics, and outbreaks. As will be explained below, the core idea of the Gillespie algorithm is that the disease state of an individual changes based on events. Each event is associated with a process, e.g., infection or recovery. These processes have rates, such that it is possible to sample the exact time of the next event from an appropriate exponentially distributed distribution. Then, at the next event time, the state of the system changes. For example, if an infection event occurs, then the number of infected individuals goes up by one and the number of susceptible individuals goes down by one. In contrast, if a recovery event occurs, then the number of infectious individuals goes down by one and the number of recovered individuals goes up by one. That's it!

Before implementing the full algorithm, it is worth discussing the algorithm in a generic form. Computer scientists often call this "pseudocode". Pseudocode is not written in a specific language but nonetheless lays out a series of steps that your actual code might follow. Here is a pseudocode version of the Gillespie algorithm for an epidemic outbreak:

```
specify time range of simulation
set initial time vector
set state vector to all susceptible except for one infectious individual
while the current time is less than the maximum
        update rate of infection based on susceptibles/infectious counts
        update rate of recovery based on infectious counts
        find the waiting time until the next event
        update the current time based on the waiting time
        decide on which process occurred
        update and store the population state based on selected event
        store the information about the system state at the current time
end
```

This pseudocode represents an ideal time to pause, strategize, and ideally discuss each step with others in your breakout room. Try to draw a sketch of what might happen given a population that has initially 4 infectious individuals vs. one in which there is initially 1 infectious individual. Is every trajectory the same? Note that in this case, both the infection rate and recovery rate depends on the state of the system. In doing so, the following schematic may be helpful. In reflecting on its meaning, also consider: what would happen if the system reached a state with 0 infectious individuals, could the outbreak still turn 'On'?



B. Definition of the Model

Stochastic realizations of the SIR model are simulated using the Gillespie framework, given the "reaction" events in the following table:

Process	**Reaction**	**rate/probability**
Infection	$S + I \to 2I$	$r_1 = \beta_I S \frac{I}{N}$
End of infectiousness (recovery)	$I \to R$	$r_2 = \gamma I = I/T_I$

These processes denote transitions amongst individuals who are susceptible (S), infectious (I), and recovered/removed (R). The total population is fixed at $N = S + I + R$. Epidemics are initiated with one infectious individual in an otherwise susceptible population. Mathematically, the initial state is $\mathbf{y} = (N_0 - 1, 1, 0)$ at $t = 0$, where the ordering of terms is susceptible, infectious and recovered. The total rate of outbreak-associated events is $r_{tot} = \sum_{i=1}^{2} r_i$. The time until next event is determined randomly such that $\delta t \sim \frac{-\log \chi}{r_{tot}}$ where χ is a uniformly distributed number between 0 and 1. In this way, the time between events follows an exponential distribution with rate r_{tot}. Then, the probability

of each event is r_i/r_{tot}. The process stops when $I(t) = 0$, as in that case $r_{tot} = 0$. After selecting an event and updating the discrete number of individuals, the reaction rates are recalculated and the process continues. The same framework can be extended to include other classes (e.g., exposed individuals) as well as other kinds of processes (e.g., post-death transmission in the case of EVD). Trajectories are complete when the epidemic dies out because there are no more infectious individuals. In the present context, we are interested in those trajectories that do not die out before the end of the simulation time.

C. Simulating a stochastic outbreak

Here, your objective is two-fold, first to modify the code below to simulate a stochastic epidemic, and then to simulate and compare results from stochastic simulations to your deterministic trajectories.

Challenge Problem: Stochastic SIR Model

Modify the following code to implement a stochastic SIR model including both transmission and recovery events in a population of size N.

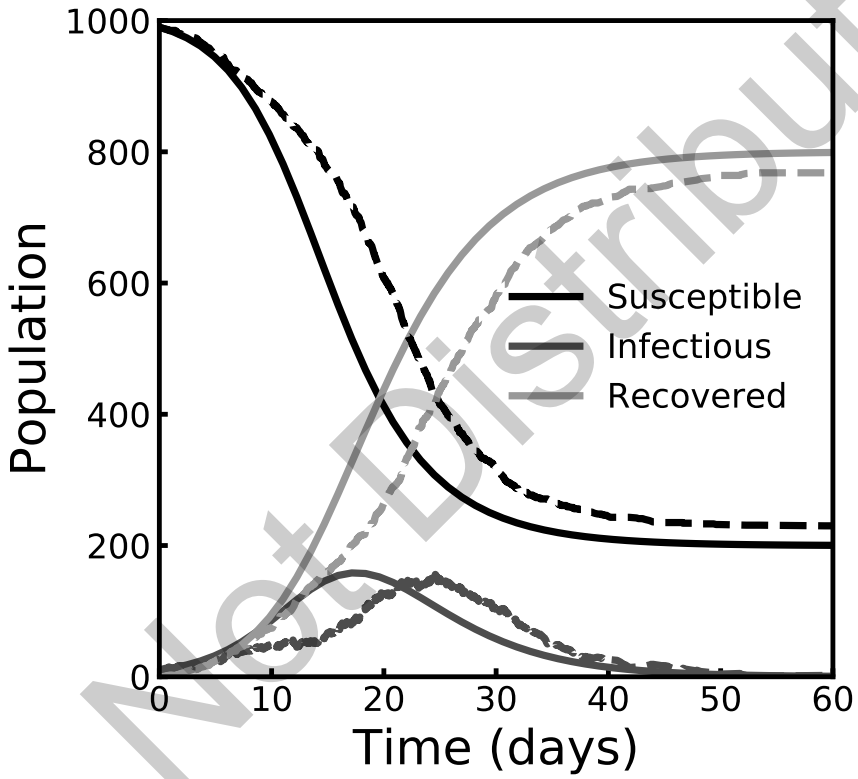
```
def stochsim_SIR(y0,trange,pars):
    """
    function [t,y] = stochsim_SIR(y0,trange,pars)

    Simulates an SIR model via the Gillespie algorithm
    from t0 to tf in trange given initial
    conditions in y0 = [S0 I0 R0] and parameters
    in pars. Returns time and values
    """

    #Conditions
    t0 = trange[0]
    tf = trange[1]
    t = [t0]
    #We will start y as a list of [S,I,R], where each triplet represents
    # a time point
    y = [y0]
    tcur = t0
    ycur = list(y0)
    ind = 0
    #Model
    while tcur<tf:
        #Check to see if there is an infection
        if ycur[1]==0:
            ind = ind+1
            t.append(tf)
            y.append(ycur)
            break
        #Rates
        infrate = ...
        recrate = ...
        totrate = infrate + recrate
        dt = -1/totrate * np.log(np.random.uniform())
        tcur = tcur+dt
        #Event type
        if np.random.uniform()<(infrate/totrate): #infection
            ...
            ...
        else: #recovery
            ...
            ...
        ind = ind+1
        t.append(tcur)
        y.append(list(ycur))
    #Now recast y as a numpy array to make it easier to work with
    y = np.array(y)
    return [t,y]
```

Once you have a working code, simulate it using the following parameters including an initial seed of 10 infectious

individuals. This is usually enough to ensure the disease spreads. The following parameters should provide both enough context and information to intialize the simulation.

```
pars['c']=20
pars['p']=0.025
pars['beta']=pars['c']*pars['p']
pars['gamma'] = 0.25
pars['tf']=60
pars['basR0']=pars['beta']/pars['gamma']
pars['N']=1000
pars['I0']=10
pars['S0']=pars['N']-pars['I0']
```

and compare directly to the SIR model which should yield the following kind of result. Here, both the stochastic and deterministic trajectories are plotted against each other.



As is apparent, there can be many differences between a deterministic model and a stochastic trajectory. Understanding that difference is at the very heart of inferring and predicting disease trajectories in real systems. If you have time, you could even try to estimate the speed of the outbreak for a set of trajectories... you will find that there is variability even without considering observational noise or process noise, and that such variation can be substantial at the outset of an outbreak when estimates are most needed!

Finally, it may seem difficult to compare the models, but it is easier than it seems. The key is to run both models but then recall that the ODE model dynamics are in terms of fractions, which must be re-scaled. Here is how to do it:

```
#Run the ODE model
t=np.linspace(0,pars['tf'])
y = integrate.odeint(sir_model,
                     np.array([pars['S0'],pars['I0'],0])/pars['N'],
                     t,
                     args=(pars,))
plt.plot(t,pars['N']*y[:,0],linewidth=3,color='b')
plt.plot(t,pars['N']*y[:,1],linewidth=3,color='r')
```

```
plt.plot(t,pars['N']*y[:,2],linewidth=3,color='g')

#Run the stochastic model
[tsim,ysim] = stochsim_SIR([pars['S0'],pars['I0'],0],[0,pars['tf']],pars)
plt.plot(tsim,ysim[:,0],linewidth=3,color='b',linestyle = '--')
plt.plot(tsim,ysim[:,1],linewidth=3,color='r',linestyle = '--')
plt.plot(tsim,ysim[:,2],linewidth=3,color='g',linestyle = '--')
```

IV. CLOSING THOUGHTS

This laboratory introduced the core concepts of epidemiological models with an eye towards informing the kind of decision making used during an infectious disease outbreak. The majority of the modules in the Foundations of Quantitative Biosciences course at Georgia Tech focus on understanding the core principles of living systems from molecules to cells to populations to ecosystems. Here, there is an important distinction. We don't want to predict a bad outcome accurately. Rather, epidemiological models are most often used to understand the likely trajectory of disease spread so as to improve critical interventions. The plural is important. As is apparent in struggling to control the ongoing COVID-19 pandemic, it is critical to develop multi-faceted, rather than a monolithic, response to epidemic outbreaks. The many facets of such a response may be unconventional, including quarantines, behavior change, the use of personal protective equipment, testing and tracing, therapeutics, and vaccination. The potential role of each of these responses can be seen by first examining simple dynamic models and then using simplified models (like the SIR or SI models) as a case study to put these ideas into practice. Putting ideas into practice often means include greater complexity (e.g., changing models into individual- or network-based frameworks). In doing so, we should also keep in mind that there are many things we don't know. This laboratory focused on one element - demographic stochasticity, the fact that individual events can lead to expected variation in the trajectory of outbreaks given otherwise equivalent pathogens and populations. Hopefully these concepts and a recognition that uncertainty is part and parcel of disease dynamics will help you in understanding, and perhaps even contributing to outbreak modeling in the years to come.