Mathematical Epidemiology

Glenn Ledder and Michelle Homp

Abstract We develop two models for the spread of an epidemic: an individual-based model and a continuous-time SEIR model. The presentation is accessible to students who have a strong background in algebra and functions, with or without calculus. Projects focus primarily on extending the models to novel situations, including more complicated disease histories and incorporation of mitigating strategies such as isolation and vaccination.

Suggested prerequisites. *Mathematical epidemiology requires strong algebra skills and a good understanding of the function concept. For maximum flexibility, each exercise, challenge problem, and research project is marked to indicate how much background is required in both calculus and computer programming. In both cases, 'level 0' means there is no requirement. Level 1 calculus means that students need to understand the derivative concept, which is presented in Section 3, while level 2 calculus indicates a requirement for some computation of derivatives and or antiderivatives. Level 1 programming means that students need to be able to run the programs presented in the appendix using either Matlab or Octave and also make minimal modifications to commands that assign values to variables. Level 2 programming means that students will need to make some modifications to the program codes in order to adapt the program to a novel scenario.*

Acknowledgements The authors wish to thank the book editors for their assistance and encouragement, Hannah Callender Highlander for suggesting this chapter, and Julie Blackwood for her thoughtful and helpful review of the manuscript.

1 Mathematical Models in Epidemiology

The reader may be familiar with the basic model of exponential growth:

Glenn Ledder
Department of Mathematics, University of Nebraska-Lincoln, e-mail: gledder@unl.edu

Michelle Homp
Department of Mathematics, University of Nebraska-Lincoln, e-mail: mhomp@unl.edu

$$P(t) = P_0 e^{kt}, \tag{1}$$

where $P(t)$ is the population of some group at time t, $P(0) = P_0$ is the initial population, and $k > 0$ is a parameter called the *rate constant* or *proportionality constant*.[1] From this example, we can tease out some details about models and modeling.

1. If you take $P_0 = 1$ and $k = 1$, the population at time 1 is $P = e \approx 2.718$, which is not an integer.
2. If you keep using larger and larger values of t, there is no limit to how large P can be.
3. The model can be rewritten in a different form by taking the logarithm of both sides to get $\ln P = \ln P_0 + kt$. This means that the graph of $\ln P$ vs t is a straight line and suggests a way to identify k from data.

Each of these statements illustrates a typical characteristic of mathematical models. First, models often give results that include all possible numbers in a range on a number line even though the quantities they represent can only take discrete values. While a value of 1079.6 is not technically correct for an integer population, there is no problem interpreting it as approximately 1080. Second, models can sometimes have qualitative properties that are not in alignment with the context they're intended to represent. Third, models can sometimes be rearranged or analyzed in ways that allow for convenient comparison with data or suggest theoretical insights.

While most of us lack the capability to collect data for growth of bacteria or other rapidly growing organisms, those who have done so will tell us that the results are never exactly the same two times in a row. The real biological world has unavoidable randomness that makes results of individual experiments unpredictable.

Although the exponential growth model is not 'correct' or 'true,' it still has value. Data from population growth experiments can adhere closely to the prediction of exponential growth until resource limits start to matter. The extreme example of this is the amount of data that can be put on a state-of-the-art computer chip, a quantity that has been growing exponentially since the 1960's and is only just starting to show signs of slowing [17].

Our epidemiological models are not going to be 'true' either. They will be based on assumptions that are oversimplifications. Nevertheless they will have value because they can make general predictions about the course of an epidemic and the impact of public health policies. For example, a model can be used to address questions such as "Will extensive contact tracing significantly decrease the negative impact of an epidemic?" Contact tracing requires significant infrastructure, so we'd like to have some idea of whether it will matter before we decide to do it. Models may not give us exact answers to questions such as these, but they are the best way we have to predict outcomes when we can't do real experiments.[2]

---

[1] You may be familiar with this model using different symbols for some of the quantities, such as $y = Ae^{kt}$. The symbols *represent* the quantities, so the model is the same no matter what symbols are used.

[2] A commonly quoted statement is that "All models are wrong, but some are useful." This statement points in the right direction, but the word "wrong" implies that the purpose of a model is to

Based of these considerations, we can make a tentative definition of the term *mathematical model*:

Definition 1. A **mathematical model** is a collection of one or more variables together with enough mathematical equations or rules to prescribe the values of those variables. Models are based on some actual or hypothetical real-world scenario and created in the hope that they will capture enough of the features of that scenario to be useful for answering research questions.

The phrase "enough of the features" requires interpretation. We can ignore minor deviations such as fractional populations or small percentage errors, but we should be on the lookout for qualitative errors. As an example, the model most often used in ecology to describe simple food web systems with one prey species and one predator species is the *Lotka-Volterra* model (see the Wikipedia article [16], for example). We can add an additional term that directly decreases the predator population through hunting. However, the model is a system of interacting components. Mathematical analysis shows that the increased predator death rate is accompanied by an increase in prey population, which in turn causes an increase in the predator birth rate. In the Lotka-Volterra model, this indirect increase fully compensates for the additional deaths, leading to the faulty conclusion that increasing the predator death rate does not actually reduce the predator population. Predator extinction from hunting is widely observed in ecology; the correct conclusion to draw from the model analysis is not that predators don't go extinct, but that the Lotka-Volterra model is fundamentally flawed [10]. Nevertheless, most treatments of the Lotka-Volterra model in books and other references fail to mention this critical flaw.[3]

Plan of the Chapter

Section 2 is devoted to an *individual-based model*, a model based on tracking the characteristics of individuals over time, much as occurs in a computer game incorporating characters with changing ability ratings and who maintain inventories of personal items.

The remainder of the chapter is devoted to *dynamical system models* that use mathematical formulas to update the total counts of people in different epidemiological states, such as *infectious* and *recovered*. The background for these models is presented Section 3. A broad classification scheme is presented in Section 4 and then the standard SEIR example is developed in Sections 5 and 6. Sections 7 and 8 discuss the various ways to obtain results from the model.

Exercises appear where needed. There are a small number of challenge problems, one that asks students to implement a model in a spreadsheet and several that ask

represent reality. The purpose of a model is to help us obtain insights about reality, so models can be "good" or "bad," but not "right" or "wrong."

[3] While the exponential growth model also fails to predict long-term qualitative behavior, it is useful in the short-term. The Lotka-Volterra model is often qualitatively inaccurate in the short term as well as the long term.

students with adequate calculus background to derive some of the important results that appear in Sections 6 and 7. Projects appear at the end of three sections: projects using individual-based models are in Section 2, while projects using dynamical system models are in Sections 7 and 8.

All of the exercises, challenge problems, and projects are classified according to the level of calculus background required and the level of programming challenge. For both, level 0 means no requirements. In calculus, level 1 means that students need the background provided in Section 3, while level 2 means that calculus computation is required. In programming, level 1 means that students need to be able to run Matlab programs (provided in the appendix) in either Matlab or Octave [12] and make minor changes to a few lines of code, while level 2 means that students have to make modest changes to the program code.

This chapter can be used in different ways, depending on the background of the students and the amount of time available for background material. Instructors who want to get to research with a minimal amount of preparation or whose students have not had calculus can focus exclusively on Section 2, which has ample scope for research projects. Instructors whose students have taken calculus may want to skip Section 2 so as to focus on the more commonly used continuous dynamical system models. Sections 7 and 8 provide complementary launch points for research, but either can be done without the other. Figure 1 shows these relationships among the sections, with one caveat: readers who wish to skip Section 7 should still read the short introduction to that section before proceeding to Section 8.

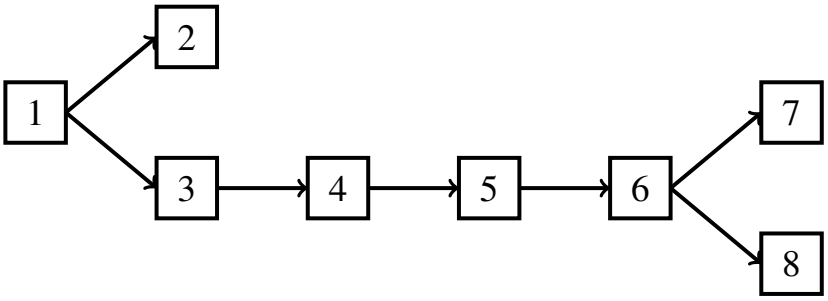

Fig. 1 Relationships among the sections

2 An Individual-Based Epidemic Model

Individual-based models,[4] or IBMs, are a good starting point for modeling because they are intuitive and can be implemented as an activity, thereby providing students

[4] The term *agent-based model* is also common.

with valuable direct experience [7]. Observing the effect of randomness inherent in IBMs helps students obtain a healthy skepticism that strengthens their ability to contextualize the results of deterministic models.

2.1 Model Description and Physical Simulation

Definition 2. An **individual-based model** consists of a database of *individuals*, each identified by one or more *attributes* that can change over time, and a set of *rules* that update the attributes of each individual at each time step.

Individual-based models can be implemented as a physical activity with actual people as the "individuals" in the model, a table-top simulation in which one modeler manages all the "individuals," or an automated simulation written with software such as Matlab or NetLogo. A description of a physical activity for our disease model is available online [6]. See [5, 13] for more about individual-based models.

For our IBM, we assume a fixed population of N individuals, each of whom has a single attribute that indicates their current epidemiological status. There are four possible states:

1. 'Healthy' individuals have not yet been infected;
2. 'Pre-symptomatic' individuals have been infected and can transmit the disease, but do not show symptoms;
3. 'Sick' individuals have been infected, can transmit the disease, and do show symptoms;
4. 'Recovered' individuals are no longer sick, cannot transmit the disease, and cannot be reinfected.

We assume individuals move linearly through these states, although that assumption could be modified as a variation of the original model. We will sometimes refer to these states using the letters H, P, S, and R.[5]

The state attribute in the model can be identified in various ways. When enacting the model as a physical activity, the participants carry a set of four colored status cards—green for healthy, yellow for presymptomatic, red for sick, and blue for recovered—and place the appropriate card on the top of their stack. In a computer implementation, the status can be a number: 0 for healthy, 1 for presymptomatic, and so on. The rules that govern the changes in states have been carefully balanced; they work best if there are at least 16 individuals in the population, with a starting point of two presymptomatic individuals in a small initial population or 2% in a large one, and the rest initially healthy.

Once the initial states have been assigned to the individuals, the simulation consists of consecutive time steps, each divided into phases:

[5] These letters are chosen to help students' intuition before we introduce formal models. In standard epidemiology models, the symbol S is used to represent susceptible individuals, but in the current context, it makes more sense to use S for sick individuals.

1. Individuals in the population are randomly assigned to pairs. This can be done using a fully random approach, such as dealing out a deck of cards (each representing a member of the population) in pairs, running a web-based app that creates pairs from a list, or using a partially random 'speed dating' structure that is more convenient for physical enactments because it takes less time [6]. In a fully automated implementation, only the healthy individuals need to be assigned a partner.
2. Healthy individuals who are paired with a presymptomatic or sick individual become infected with probability 5/6; in a physical enactment, this is easily done by rolling a die and equating a die roll of one with avoiding the infection. Those who do become infected advance from healthy to presymptomatic.
3. After all pairs have been checked for disease spread, any individuals who were presymptomatic at the beginning of the time step become sick, while those who were sick at the beginning of the time step become recovered.
4. The numbers of individuals in each status category are recorded for subsequent analysis.

The simulation proceeds from one day to the next, each day starting with new partner assignments and continuing with status updates. Eventually the simulation ends when all remaining individuals are either healthy or recovered. Once there are no presymptomatic or sick individuals remaining, there can be no further status changes.

This simple individual-based model captures many of the features of real disease spread. Individuals interact randomly and possible transmission encounters may or may not result in actual transmission. Other features are not so realistic. In reality, the amount of time spent in a presymptomatic or sick state can vary from one person to another. Most diseases have an incubation period of noticeable duration, necessitating a latent stage between the healthy and presymptomatic stages. In real disease settings, individuals have large numbers of daily contacts, each with a low probability of transmission. Our individual-based model obtains realistic results with a simple mechanic by reversing this pattern, with only one daily contact and a high probability of transmission.

The simplicity of this individual-based model also highlights the critical role assumptions play in modeling a phenomenon. In this context, the need for and outcomes of changing the model's assumptions can easily be understood, demonstrating the importance of consulting with experts to ensure a model incorporates critical features and exhibits necessary behaviors. (Some of these assumptions will be considered in the exercises.)

2.2 Computer Simulation

While a physical simulation is a great way to build intuition, it is not a great way to study general model behavior. Instead, it is helpful to write a computer simulation for our individual-based model. We have provided two resources for this purpose.

Challenge Problem 1 provides detailed directions for creating a spreadsheet implementation of the HPSR individual-based model. This can be used (with modification) for the research projects in this section.

Spreadsheets have the advantage of familiarity and an intuitive structure. They are not generally used for serious modeling work because of two significant drawbacks: (1) the formulas that encode the model can only be viewed singly rather than as a package, and (2) they lack some essential programming functionality, such as the capacity to easily repeat computations with minor variations.

As an alternative to spreadsheet programming, we recommend that students use computer programs written for the Matlab programming environment. The Matlab environment has its own language, not much different from the better-known Visual Basic language, along with a large number of powerful, built-in functions that do complicated computations. Since few students who take mathematics courses at any level have any programming background, we provide Matlab programs we wrote specifically to be accessible to beginners.

The Appendix contains a suite of three Matlab programs that automate the model computations and obtain results from the output data. We present here a brief description of these programs. Most academic institutions have agreements that allow students to obtain Matlab at little or no cost. For those that don't, there is free online software called Octave that is able to run most Matlab programs [12].[6] We assume that students have acquired basic operational knowledge of how to load programs into the edit windows and how to run the programs from those windows.

Matlab programs are of two types: scripts and functions. Scripts are self-contained lists of instructions written in syntax that can be understood by an interpreter that translates the instructions into computer code. These can be run from the Matlab editor or an Octave window by clicking the *Run* menu item.[7]

Functions are special programs that automate some of the more complicated program tasks. These have a specific name along with a list of arguments and output variables. Part of the strength of Matlab as an environment for scientific computation is its extensive library of built-in functions, while another part of its strength is its facility for user-defined functions.

Functions cannot be run as separate programs, but are instead called from a command window or run from a script, just as one might use *y=cos(pi)*. The program suite for the IBM is based on a function *[H,P,S,R]=hpsr(b,N,P0,S0)*,[8] which au-

[6] As of this writing, the programs all run successfully in Octave.

[7] In Matlab, any changes entered in a script will be saved before the script is run. In Octave, you must manually save changes before rerunning a script.

[8] By convention, we choose to give functions names that start with a lower-case letter, while script names start with a capital letter.

tomates the entire individual-based model computation and returns the time series of class counts. Function *hpsr* requires input values for the transmission probability (*b*), the total population (*N*), and the initial numbers of presymptomatic (*P0*) and sick (*S0*). It needs to be modified for Programming Level 2 projects to match changes in the model rules, but does not need to be changed for Level 1 projects.

The IBM program suite includes two scripts that make use of function *hpsr*.

- *HPSR_sim.m* uses *hpsr* to run one simulation and produce a plot of the results. The user needs to modify the *SCENARIO DATA* section for each run. The function call in the *COMPUTATION* section and some of the *OUTPUT* instructions will need to be changed for Level 2 projects.
- *HPSR_avg.m* collects data from multiple simulations and computes the mean and standard deviation of two key results: the maximum infected fraction and the final healthy fraction. The user needs to modify the *SCENARIO DATA* section for each run. The function call in the *COMPUTATION* section will need to be changed for Level 2 projects.

Section 2 Exercises

Exercise 1. (Calculus Level 0, Programming Level 0)
Run the individual-based model three times using either a physical simulation, with people playing the roles of the individuals, or a table-top simulation. In each case, plot graphs of the daily class counts. Also record the number and identities of individuals who remained healthy throughout the simulation. Are those who don't get sick different in any meaningful way from those who do? In a real situation where an illness is spreading among a population, might there be a difference between those individuals who don't get sick and those who do? Discuss.

Exercise 2. (Calculus Level 0, Programming Level 0)
The individual-based model uses four different states, each marked by a different color in the physical simulation, to describe the illness progression.

(a) Using the rules as described, explain which, if any, of the color distinctions are needed and which are not.
(b) Now suppose the rules for the physical simulation are changed so that half of the individuals who are sick choose to isolate. Explain which, if any, of the color distinctions are needed and which are not.
(c) How might the rules change if some of the individuals are vaccinated? How might this impact the meaning or number of the colored status cards?
(d) How might the rules change if individuals could contract the same illness more than once? How might this impact the meaning or number of the colored status cards?

Exercise 3. (Calculus Level 0, Programming Level 0)
The individual-based model described in this unit assumes the duration of each

phase of the illness is a single day. In reality, most illnesses do not progress this quickly. Visit the website [3] for information about the H1N1 virus, often referred to as "swine flu." Then develop a set of status cards and rules for an individual-based model of H1N1. Assume that the model would incorporate isolation of sick people. Would you still continue to use the same four states or would you need more? How many status cards would you use for each state? How many would you need for the full simulation? Would you need to incorporate additional die rolls into the model? Fully justify any assumptions made about the states of the illness and the duration of each state.

Exercise 4. (Calculus Level 0, Programming Level 1)
Run the individual-based model program *HPSR_onesim* three times using parameter values that match your physical simulation in Exercise 1. Compare the results. Do they give convincing evidence that the computer simulation and physical simulation actually implement the same model?

Exercise 5. (Calculus Level 0, Programming Level 1)
Run the individual-based model program *HPSR_avg* using parameter values that match your physical simulation in Exercise 1. Compare the results. Is this a good way to check that the computer simulation and physical simulation are actually implementing the same model?

Section 2 Challenge Problem

Challenge Problem 1. (Calculus Level 0, Programming Level 0)
Create a spreadsheet implementation of a slightly modified version of the individual-based model described in the section. Here are some recommended guidelines:

1. Use row 1 to list the four parameters by symbol: the transmission probability b, the population size N, and the initial counts of presymptomatic P_0 and sick S_0. The values of these parameters go in row 2, beneath the parameter names. For a first run, We recommend a transmission probability of 5/6 or 1 and a population of 1000 that includes 20 presymptomatic and 0 sick.
2. Use row 4 to enter headings for the data columns. Columns for the day and the four population classes (H, P, S, R) are a minimal requirement.
3. We recommend having three additional columns for the total population, the infectious population $I = P + S$, and the number of new infections.
   a. The values in the total population column should be calculated using a formula that sums the four classes. This serves as a check on the spreadsheet formulas.
   b. The values in the infectious column are just the sums of the P and S columns. The main benefit of this column is that a graph of the daily infectious count is very useful.

   c. The new infections column is helpful because it is good to do calculations in several small formulas rather than one messy one. Without identifying individual pairings, we can estimate the fraction of healthy people whose "partner" in the simulation is infectious as the fraction of the total population that is infectious. Then we can assume that a fraction b of these healthy people with infectious partners become new infections.

4. Each of the columns needs to be given formulas to determine the values using the data in row 2. We recommend running the simulation for 30 days. The initial counts for classes H, P, S, and R can be calculated from the values entered in the cells for N, P_0, and S_0. Subsequent values in these columns can be calculated from the previous values along with the number of new infections in the last column.
5. Spreadsheets do computer calculations, which means that they produce numbers with many decimal digits. This makes it very hard to read the results. We recommend that you format all of the cells in the data table as "Number" with 0 decimal digits.
6. Raw numbers are hard to interpret. Your spreadsheet will be more useful if you make a graph. Use the scatter plot with only points because the data is discrete. Plot three data series: H, I, and R. Make sure your graph has axis labels and a legend. You can use the default color and style scheme; alternatively, you can match the color scheme of the card system by using dark green for Healthy, red for Infectious, and blue for Recovered. The graph will be easier to read if you use different styles for the points in each data series. One possibility is shown in the Matlab program *HPSR_onesim.m*.

When you are finished, try using parameter values that match your physical simulation so that you can make sure the results are consistent. Alternatively, use a comparison with *HPSR_avg* with the recommended population size of 1000.

Section 2 Projects

In each of these projects, it is best to start by using the physical simulation to make sure that your model changes make sense. Then use the spreadsheet from Challenge Problem 1 and/or the Matlab program suite to collect data.

Research Project 1. (Calculus Level 0, Programming Level 0 or 1)
Study the effect of the transmission probability b on the results of the disease in the individual-based model. Hint: You can collect useful data for any value of b using the spreadsheet implementation of Challenge Problem 1 or by running *HPSR_avg*.

Research Project 2. (Calculus Level 0, Programming Level 0 or 2)
Suppose patients with the disease are sick for two days instead of one. Modify the model to account for that possibility. Try keeping the transmission probability the same and also adjusting it downward by some amount. Determine what transmission probability yields results most like those of the original model with the larger transmission probability and one day of sick time. Hint: You can break the sick class up into two subgroups.

Research Project 3. (Calculus Level 0, Programming Level 0 or 2)
As an extension of Research Project 2, suppose a fraction p of patients with the disease are sick for two days, while the remainder are sick for just one day. Design experiments with this model and describe and explain the results.

Research Project 4. (Calculus Level 0, Programming Level 0 or 2)
Suppose individuals have two contacts per day instead of one. Modify the model and compare the results with the original model.

Research Project 5. (Calculus Level 0, Programming Level 0 or 2)
Add isolation of the sick to the original model or one of the variations described in other projects. Note that you will need a parameter to represent the probability that an individual isolates when sick. The effect of isolation will depend on the value of this parameter.

3 Continuous Time Dynamical Systems

While the individual-based model of Section 2 is discrete, most epidemiology models take the form of continuous-time dynamical systems. These systems consist of differential equations, which means that they are based on calculus. However, some of the background needed to understand these systems is not part of the standard

calculus curriculum; conversely, very little of what does appear in a calculus curriculum is actually needed for our purposes. The goal of this section is to provide a self-contained introduction to dynamical systems that can be understood by students with a strong background in algebra and pre-calculus.

3.1 The Derivative

Standard first courses in calculus focus on the *derivative*. The text material includes the motivation for the derivative, its definition, rules for computing derivatives, and a variety of applications of the derivative. While all of this material is important in calculus, all we really need for dynamical systems is the geometric concept of the derivative. Here is a concise summary.[9]

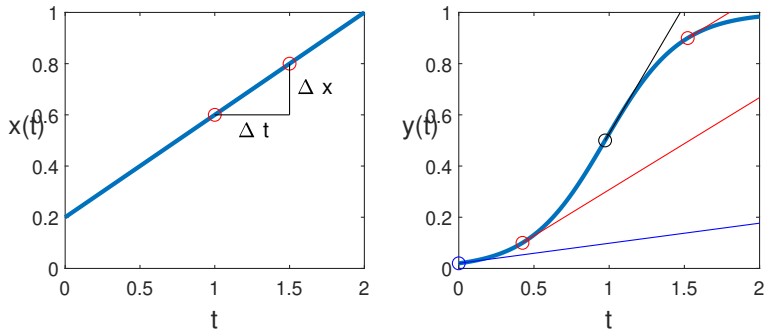

Fig. 2 A linear function $x(t)$ and a nonlinear function $y(t)$ (the thick curves), showing the slope and instantaneous slope (the thin lines) of the two functions at various values of t.

Figure 2 illustrates a pair of functions, $x(t)$ and $y(t)$, that might represent the population size of a growing community. The function x shows *linear* growth, as would happen when individuals join the community at a fixed rate from the outside. That rate is the change in x divided by the change in t, which is the slope of the line in the graph. We can calculate the slope using algebra by picking two points, calculating the differences in their x and t values, and taking the ratio; here,

$$\text{slope} = \frac{\Delta x}{\Delta t} = \frac{0.8-0.6}{1.5-1} = 0.4. \tag{2}$$

The function y shows what is called *logistic* growth, which means that the growth is limited by the availability of resources. In this instance, the population grows by virtue of having more births than deaths, with no migration. It grows slowly at first

[9] See [9] for a more complete presentation.

because there are few potential parents, then increases more rapidly, and then levels off as the availability of resources or space becomes critical.

It still makes sense to talk about the slope of the graph of y, which we can think of as the slope of the line tangent to the graph at a particular point. A tangent line has only one known point, so we can't calculate its slope with standard algebra; techniques for determining the slope of these tangent lines are developed in calculus. For our purposes, we can assign a symbol to represent the slope at a point t and focus on its interpretation. The symbol we use is dy/dt, which is similar to the notation $\Delta y/\Delta t$ for the slope of a line. The key difference is that we have to interpret dy/dt as a single symbol, not a quotient of two different things.

Definition 3. For a function $y(t)$ that has a smooth graph, the **derivative** dy/dt at any given point is the slope of the tangent line at that point, which represents the instantaneous rate of change of the quantity y.

3.2 Dynamical Systems

In a calculus text, the derivative of a function $y(t)$ is nearly always given as a function of the independent variable t. In mathematical modeling, we often have systems in which the rates of change are dependent only on the state of the system, which means that the derivative is given as a function of the dependent variables.

Definition 4. An **autonomous dynamical system** is a system of quantities whose rates of change in time are given as functions of the quantities themselves.

As an example, the dynamical system model

$$\frac{dy}{dt} = f(y) = 4y(1-y), \qquad y(0) = y_0, \qquad 0 < y_0 < 1 \tag{3}$$

where y is a function of t, represents the spread of misinformation (which can be considered as similar to the spread of a disease). Everyone is either 'susceptible' to the misinformation or 'infectious' with it, and there is no 'recovery'. The dependent variable y is the fraction of the population that is infectious. This fraction increases for any current fraction y between 0 and 1 according to the function $f(y) = 4y(1-y)$.

How should we try to understand the dynamical system (3)? The simplest way is to use the information shown by a graph of the known function $f(y) = 4y(1-y)$ for the range of values from $y = 0$ to $y = 1$, not to find a formula for the function $y(t)$, but to check slopes on a sketch of the graph of y vs t. If we know the current value of y at any point in time, then we can use the graph of f to identify dy/dt, which is the slope of the graph of $y(t)$ at that particular time.

Figure 3 shows two graphs, one of the function $f(y)$ and one of the function $y(t)$. The first graph gives us information we can use to verify the second graph. Suppose the value of y is initially 0.01. From the left panel of Figure 3, we can see that the value of f, and hence of dy/dt, is small and positive. So y is increasing, but with

a fairly flat rise. The first marked point in the right panel shows the same value of y and a tangent line that has the slope found from the f axis in the left panel. As y increases, the graph of f tells us that y will continue to rise with increasing slope. In other words, it will curve upward as it increases. The second marked point is at $y = 0.1$ and shows the value of dy/dt to be a little less than 0.4. The corresponding point in the panel on the right has $y = 0.1$ and shows the tangent line with the correct slope. This behavior starts to change when we reach $y = 0.5$, since that is where f achieves its maximum value. This point corresponds to the steepest slope on the graph of $y(t)$. After that, the function f is still positive, but its value decreases as y increases further. The graph of $y(t)$ continues to increase, but with a flattening, or downward curvature. The fourth marked point has the same value of f as the second, which means that the slopes of the graph of y at those two points are the same. Note that the graph of $y(t)$ can never cross the threshold value $y = 1$ because f is 0 when $y = 1$, meaning that y is no longer increasing.

The graph of y in the right panel was created by applying calculus to obtain the solution formula for the system (3); however, it could instead have been obtained from a numerical simulation using a method presented in Section 8. Without one of these methods, we would not have been able to get the times right for the four marked points; however, we would still have been able to get the right slopes for each value of y, which give us the correct overall shape for the graph.

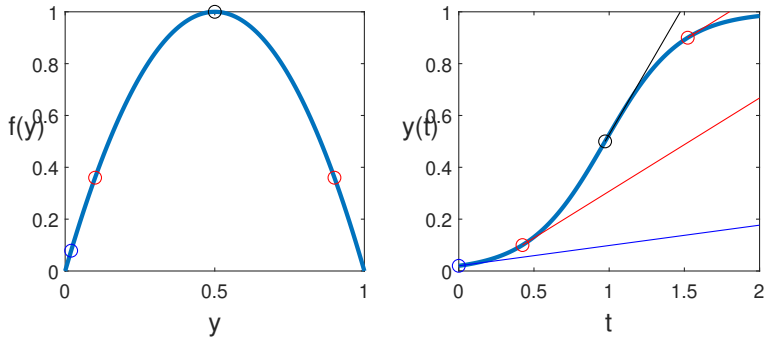

Fig. 3 The function $f(y) = 4y(1-y)$ and the solution of $dy/dt = f(y)$ with $y(0) = 0.01$

To summarize: a differential equation of the form $dy/dt = f(y)$ specifies the slope of the graph of y as a function of the value of y. This information, along with an initial condition, is sufficient to define the graph, which we can roughly sketch with nothing more than logical statements about how f changes as y changes.

It is not a large leap to appreciate that this same idea will work for a system of four differential equations for variables S, E, I, and R, where the derivatives are known functions of the four variables. The rates of change for each variable will be functions of more than one state variable, so we won't have a graph like the one

in the left panel of Figure 3; however, we will have the formulas that calculate the slopes from the state of the system.

Section 3 Exercises

Exercise 6. (Calculus Level 1, Programming Level 0)
The differential equation $dy/dt = ky$, where k is any positive constant, models exponential growth. In contrast, explain why the equation (3) and the graph of its solution (see the graph on the right in Figure (3)) is an appropriate model for a scenario in which there is a constant but limited amount of available resources.

Exercise 7. (Calculus Level 1, Programming Level 0)
As noted in Exercise (6), the differential equation in (3) is used to model the growth of a population with a fixed resource limitation. In models representing the interaction between predator and prey populations, the function $y(t)$ can represent the predator while the prey can be considered the limited resource. For what situations would the equation in (3) be appropriate for modeling the population of a predator species for a given prey? For what predator-prey situations would it *not* be appropriate? Justify your claims.

Exercise 8. (Calculus Level 1, Programming Level 0)
The dynamical system model

$$\frac{dy}{dt} = f(y) = 4y(1-y) - 3y, \quad y(0) = 0.01$$

represents the infectious population fraction for a disease in which recovered individuals can be reinfected (like the common cold). Plot the graph of f and use it to sketch a possible graph for $y(t)$. (Keep in mind that y represents a population fraction.) Check your graph by plotting the function

$$y(t) = \left(4 + 96e^{-t}\right)^{-1},$$

which is the solution of the differential equation problem.

Exercise 9. (Calculus Level 1, Programming Level 0)
Repeat Exercise 8, but with $y(0) = 0.5$. [Keep in mind that you won't be able to use the given solution formula as a check.]

Exercise 10. (Calculus Level 2, Programming Level 0)
Show that the function

$$y(t) = \left(4 + 96e^{-t}\right)^{-1},$$

solves the initial value problem

$$\frac{dy}{dt} = f(y) = 4y(1-y) - 3y = y - 4y^2, \quad y(0) = 0.01$$

Hint: If we define $u(t) = 4 + 96e^{-t} = y^{-1}$, the right side of the differential equation becomes $(uy)y - 4y^2 = (u-4)y^2$.

Exercise 11. (Calculus Level 2, Programming Level 0)
Show that the function

$$y(t) = \left[\frac{b}{a} + \left(\frac{1}{y_0} - \frac{b}{a}\right)e^{-at}\right]^{-1}$$

solves the initial value problem

$$\frac{dy}{dt} = ay - by^2, \quad y(0) = y_0.$$

Hint: See Exercise 10.

4 Dynamical System Models

Model development is as much an art as it is a science. The science is in the employment of logical structures to serve as the model framework, while the art is in finding creative ways to conceptualize the motivating real-world scenario.

In Section 2 we saw one class of models—individual-based models. They have the advantage of easy conceptualization and inclusion of randomness, but they have some drawbacks as well:

1. They have to be run many times to get a picture of the expected behavior.
2. They become very complicated when more realistic features are added, such as random durations of incubation and infectivity.
3. They do not lend themselves to mathematical analysis.

Most mathematical modeling in epidemiology instead uses dynamical systems models, which we explore in the remainder of the chapter. Dynamical system models in epidemiology have a long history [2]. The class of models most commonly used today was first introduced in a seminal paper by W.O. Kermack and A.G. McKendrick published in 1927 [8].

Dynamical system models are based on the same principle as accounting. If you want to know how much money you have in different budget categories, you can (1) keep the money in different boxes and count it every day, or (2) keep track of your income, expenditures, and internal transfers, and use these amounts to calculate daily totals. In epidemiology, the budget categories are replaced by variables that indicate the current totals of people in various epidemiological states, such as 'Susceptible' (which we called 'healthy' in the individual-based model) or 'Infectious' (which often includes both presymptomatic and sick individuals).

Mathematical epidemiology is a rich subject for which one can find books at levels from advanced undergraduate to research level.[10]

4.1 Classification of Dynamical System Models

Several features are needed to classify epidemiology models. We present these in rough order of importance.

1. Disease Type

   Infectious diseases can be divided into two subgroups, based on the transmission mechanism.

   - **Person-to-person** transmission involves either direct transmission through physical contact or indirect transmission through the environment, such as droplets that enter the air through sneezing or coughing.
   - **Vector-borne** transmission is required for diseases in which the pathogen has a complicated life cycle that requires multiple host species. An example is malaria, which is caused by a protozoan that lives part of its life in humans and part in mosquitoes.

   Most infectious diseases which affect humans fit neatly into one of these two categories, but some are less clear. The plague that swept through much of the world in the medieval period had two forms: bubonic, which humans contracted from fleas, and pneumonic, which was passed from person to person via droplets in the air.

   In this chapter, we consider only diseases with person-to-person transmission.

2. Time Frame

   - **Epidemic** models are designed for the short term. Their distinguishing characteristic is that there are no mechanisms for replenishment of susceptible people. Like a forest fire that must eventually run out of fuel, an epidemic scenario necessarily ends with no remaining disease. Study of epidemic models focuses on rate of spread, change in the number of infected individuals over time, and final susceptible population size. Analytical methods are available for some results in the simplest cases, but usually simulations are required. Epidemic models can be either deterministic (no randomness) or stochastic (having some random elements), depending on the level of detail desired for quantifying the various disease processes.

[10] Space does not permit an exhaustive list of references. For lower-division undergraduates, we recommend the oddly-titled book by Smith [15] because it builds up to a sophisticated level from a very humble beginning. For advanced undergraduates, the paper by Blackwood and Childs [1] is a good place to start.

- **Endemic** models are designed for the long term. They always include at least one mechanism for replenishment of susceptibles, typically birth of susceptible individuals, but there can also be processes such as immigration or gradual loss of immunity. There are also diseases, such as the common cold, that do not confer subsequent immunity at all, perhaps because there is a plethora of strains or very rapid mutation. The long-term behavior of endemic models can be obtained using computational and analytical methods.

Many treatments of mathematical models in epidemiology focus primarily on endemic models, because these are the ones for which mathematical analysis can make the strongest conclusions. In this chapter, we instead focus on epidemic models to reflect their importance in the COVID-19 dominated world of 2020 and beyond.

3. Population Constancy

The total population can be fixed by omitting demographic processes altogether, by making sure birth and death rates are equal with no additional mortality or migration, or, in the case of epidemic models, counting deceased individuals as equivalent to recovered ones. This simplifies the model as compared to the more common case of variable size populations and is typically done for all epidemic models as well as some endemic models.

4. Classes

Although the choice of classes is no more fundamental than the choice of time frame, it is traditional to name models according to the list of classes used in them. Each class tracks the number of individuals with that particular disease status. Our starting point is the SEIR model, where

- S is *Susceptible*, for individuals who are at risk of catching the disease. (We called this group "Healthy" in the individual-based model.)
- E is *Exposed*, for individuals who have been infected but are not yet infectious. This is a poor choice of term–'latent' is more accurate. In everyday language, we would say that a person has been 'exposed' when they have contacted an infectious person, regardless of whether they have caught the infection, but in epidemiology all members of the 'Exposed' class have been infected. As a compromise between accuracy and consistency, we will retain the class symbol E but use the term *Latent* as the class name. (The individual-based model did not have a latent class.)
- I is *Infectious*, for individuals who can transmit the disease. In the basic SEIR model, infectious individuals may or may not have symptoms. (In the individual-based model, the infectious class was divided into presymptomatic and symptomatic subgroups.)
- R is *Removed*, for individuals who are not currently infectious and are immune from further infection. This can include individuals who have recovered

and individuals who are still sick but no longer infectious. Including deceased individuals as 'removed' is convenient if death by the disease is the only demographic process, as then the population can still be thought of as 'constant' in the sense that the class system is then closed.

Many disease models include additional or different classes based on features of the particular disease being studied. There could be isolated infectives[11] and quarantined susceptibles, and infectives could be further subdivided into asymptomatic and symptomatic. A COVID-19 model should distinguish between asymptomatic and symptomatic infectives and between unconfirmed, confirmed, and hospitalized cases.

5. Processes

In addition to a list of epidemiological classes, a model needs a list of processes that cause individuals to move from one class to another. In an SEIR model, there must be a transmission process that moves susceptible individuals into the latent class, an incubation process that moves latent individuals into the removed class, and a removal process that moves infectious individuals into the removed class. There could be other processes, such as vaccination, isolation, and loss of immunity. Some of these are considered in the projects.

6. Dynamical System Type

Discrete-time models are based on algebra, while continuous-time models are based on calculus. Discrete-time models can seem more intuitive, but continuous dynamical systems have much better mathematical properties. We will follow the more common practice of using continuous models.

5 Building the SEIR Epidemic Model

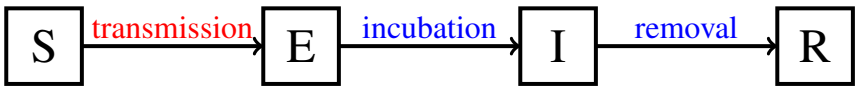

Fig. 4 The SEIR epidemic model in words.

Figure 4 shows a schematic of the structure of the standard SEIR model. The four population classes are related by three processes:

1. A **transmission** process moves susceptible individuals into the latent class (E).

[11] Technically speaking, the word 'infective' does not appear in official dictionaries; however, it is a much better term than the alternative 'infectious individual.'

2. An **incubation** process moves latent individuals into the infectious class.
3. A **removal** process moves infectious individuals into the removed class.

Notice what is missing:

- We have not explicitly included disease-induced mortality, but that does not mean we haven't allowed for it. It is customary in epidemic models to make no distinction between people who have died from the disease and people who have recovered. This undoubtedly seems horrible to the novice modeler, but it is actually very instructive. From a *human* perspective, we want to distinguish between healthy people, sick people, recovered people, people with ongoing deleterious effects, and people who died. But our goal is to make an *epidemiological* model. There is no epidemiological distinction between infectious people who are sick and those who are not, or between removed people who are healthy, still suffering illness, or deceased. These human interest features can be added to the base epidemiological model during analysis.
- We have no birth or natural death processes and no migration. Obviously these demographic processes do occur during a disease outbreak. However, on an epidemic time scale of weeks or months, births and natural deaths only change the class counts to a limited extent. Remember that our model is not intended to exactly match reality. We will see assumptions in the quantification of the epidemiological processes that will yield more quantitative error. Some of these will be explored as research projects.
- We have not included any modifications for public health measures, such as vaccination, isolation of infectious individuals, or quarantine of individuals through contact tracing. The end-of-chapter projects will focus largely on adding possible modifications. There are many ways to do this, not all of which have been thoroughly explored. This is part of the art of mathematical modeling, and it offers students an opportunity to do original research by being the first to think of a new idea.

From the schematic diagram, we can immediately write a set of conceptual equations for the changes in the class sizes. The susceptible population is affected by only one process–the transmission process, which decreases the population:

$$\frac{dS}{dt} = -\text{ transmission rate.} \tag{4}$$

Similar equations describe the overall rates of change for the other classes:

$$\frac{dE}{dt} = \text{ transmission rate} - \text{ incubation rate,} \tag{5}$$

$$\frac{dI}{dt} = \text{ incubation rate} - \text{ removal rate,} \tag{6}$$

$$\frac{dR}{dt} = \text{ removal rate.} \tag{7}$$

Now suppose we define N to be the total population; that is,

$$N = S + E + I + R. \tag{8}$$

The change in N is the sum of the changes in the four components. Since each process increases one class at the expense of another, the overall population stays the same (given the interpretation of 'deceased' as being epidemiologically equivalent to 'recovered'). This means that we could omit one of the change equations; for example, we could omit the R equation and instead use the N equation to calculate R. This is commonly done when focusing on the mathematical analysis of a model. However, when doing a computer simulation, it is good practice to use the change equations to calculate all class counts and then use the N equation as a check for possible error in the program.

While we could use N to represent the actual population, it is usually best to take $N = 1$, which means that the class sizes are used to measure fractions of the total population rather than raw numbers. Fractions of a total are meaningful with no additional context, whereas class counts are only meaningful in comparison to the total population.

5.1 Quantifying the Processes

To complete the SEIR model, we need to find mathematical formulas for each of the three processes in the compartment diagram and conceptual equations. There are standard choices that are used in nearly all circumstances. These standard choices are based on some questionable assumptions, so it is important to understand the formulas in some detail. (See [15] for a similar discussion with much more detail.)

The first step in quantifying each of the processes is deciding what variables must be considered. Take removal as an example. If a lot of people are infectious now, then surely people will leave the infectious class at a high rate; however, if nobody is infectious, then nobody can be removed from the infectious class and the removal rate is 0. Thus it is reasonable to assume the removal rate depends on the size of the infectious class. What about the other three classes? Does the amount of removal from the infectious class depend on how many people are susceptible, how many are latent, or how many are already removed? The answer is 'no' in all cases. Whether an infectious person is removed in a given day depends only on what is happening inside that infectious person, not on the states or changes in state of anyone else. The removal process is an example of a *transition*: a process that happens to each individual person in the departing class, independent of anyone else. Similarly, the incubation process is a transition. Each latent individual becomes infectious based on their own internal changes, independent of what is happening to others. The more people in the class that undergoes the transition, the greater the rate at which the transition occurs at the population level.

The transmission process is fundamentally different from the transition processes. Susceptibles are necessary for transmission to occur. But unlike incubation, which happens to latent people without any interaction, transmission doesn't happen independently. You can't get the disease without getting it from someone who is infectious.[12] Thus, the formula for the transmission process is going to need to use both the susceptible and infectious class counts. Accordingly, we consider transitions and transmission separately.

Transition Processes

Let's start with the incubation process. If we compare two moments in the history of an outbreak, and there are twice as many latent people in moment A than in moment B, it seems reasonable that the rate of incubation is twice as great at moment A. We therefore assume that the incubation rate is proportional to the latent class size.[13] This assumption tells us the mathematical form of the rate formula, but with a proportionality rate constant that is specific for each disease. It is customary to use lower-case Greek letters for this and other rate constants. The specific rate constant symbol for a particular process varies from one author to another. We'll use the Greek letter 'eta' for the incubation rate constant. Thus, we have

$$\text{incubation rate } = \eta E. \tag{9}$$

The removal process is analogous to the incubation process in that its rate is proportional to the size of the class it is leaving. Using the Greek letter 'gamma', we have

$$\text{removal rate } = \gamma I. \tag{10}$$

So far this may seem straightforward, but there is actually a lot more that needs to be said. As a thought experiment, suppose that the entire latent class is made up of people who just got infected yesterday. Assuming a mean incubation time of 5 days, the proportionality assumption seems to say that 1/5 of these people will progress to the infectious class today. This is a very suspicious claim. If it takes an average of 5 days, then most people will become infectious in 4, 5, or 6 days. It seems likely that nobody would become infectious in the first day, as that would require the disease to progress 5 times as fast as the average. Similarly, if nearly everyone in the latent class has already been infected for 5 days, then it seems likely that more than 1/5 of these overdue patients will become infectious in the next day. The naive assumption that the incubation rate constant for latent patients will be the same each day is a significant conceptual error, far worse than the minor error of neglecting the small

[12] Some diseases are spread through the environment, but the disease agent gets into the environment from infectious individuals. Epidemiologically, it doesn't matter if you get the disease by shaking hands with someone or by breathing in disease agents the other person left behind when leaving a room.

[13] This can be shown to be equivalent to the assumption that the incubation times are exponentially distributed, with the rate constant given by the reciprocal of the mean time.

number of births that occur during the scenario. The reality is far more complicated than what we are assuming in our model.

In spite of the apparent seriousness of this conceptual error, there is some reason to think that it won't actually matter. Most of the time, the latent class will consist of people who were infected at different times. If our naive model predicts that someone who is newly infected suddenly becomes infectious today, while someone who has been infected for 6 days already does not, these errors should average out at the population level. This is really what we are assuming. While this issue is largely ignored in epidemiological modeling, we should keep it in mind whenever we are tempted to take our results too seriously. Exploration of this issue makes a nice research project (given here as Research Project 8, with background from Exercise 13).

The Transmission Process

Suppose each infectious person encounters a fraction c of a population of size N per day. Assuming that all possible encounters are equally likely, we can expect that a fraction S/N of these encounters are with susceptible individuals. This means that each infectious person has cS encounters with susceptible individuals. If each encounter has a probability p of resulting in a transmission, then we can expect one infectious person to transmit the disease to pcS individuals per day. If we have I infectious individuals rather than 1, the total rate of transmission is $pcSI$. We don't need individual factors p and c, so we can write the transmission rate as

$$\text{transmission rate } = \beta SI, \tag{11}$$

where β is the product of the contact fraction and the transmission probability. This formula can be used regardless of whether we are measuring the population classes as counts of people or fractions of the total population, but the value for β is different in the two cases. This is one of several reasons why it is hard to identify a value for this parameter. This issue will be addressed later.

As with the transition formulas, there are hidden assumptions in the simple formula for the transmission rate. If you reread the explanation, you should notice that it doesn't distinguish individuals from each other. In reality, we live in a social network where we have frequent contacts with some people and no contact with nearly everyone else. Some of us are at the center of our network and have lots of contacts, while others have far fewer contacts. In the COVID-19 pandemic, some people are deliberately reducing the number and frequency of contacts, while others are not. Additionally, infectious individuals vary in their level of infectivity and susceptibles vary in their level of susceptibility. As with transition processes, we are assuming that we can use averages. On the other hand, it was easy to come up with a thought experiment in which the transition formula gives a value that is too high or too low, even averaged over the whole population at any given time. It is much harder to do that for the transmission process. While there may be some error introduced in

an epidemiology model by the overly simple assumptions about transmission, it is almost certainly less error than is introduced by the assumptions about transitions.

5.2 The Final Model

Now that we have chosen formulas for the rates of the processes, it is helpful to redraw the compartment diagram, using those rates as the labels on the arrows (Figure 5).

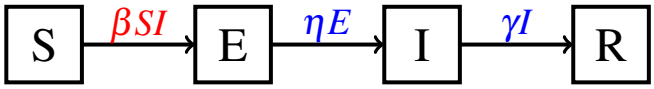

Fig. 5 The SEIR epidemic model in symbols.

Working directly from the diagram gives us the differential equations that describe the rates of change in terms of the state of the system. Full model specification also requires initial conditions. We'll use lower-case letters for these, as R_0 is often used for something entirely different from the initial condition parameters. The final model is

$$\frac{dS}{dt} = -\beta SI, \qquad S(0) = s_0 > 0; \tag{12}$$

$$\frac{dE}{dt} = \beta SI - \eta E, \qquad E(0) = e_0 \geq 0; \tag{13}$$

$$\frac{dI}{dt} = \eta E - \gamma I, \qquad I(0) = i_0 \geq 0; \tag{14}$$

$$\frac{dR}{dt} = \gamma I, \qquad R(0) = r_0 \geq 0; \tag{15}$$

where

$$s_0 + e_0 + i_0 + r_0 = 1, \qquad e_0 + i_0 > 0. \tag{16}$$

These last two requirements ensure that the initial population total is 1 and that there are some infected people to get the outbreak started.

Section 5 Exercises

Exercise 12. (Calculus Level 1, Programming Level 0)
Construct a compartment diagram similar to that of Figure 5 for the class and process structure that matches the individual-based model of Section 2.

Exercise 13. (Calculus Level 0, Programming Level 0)
Suppose there are two latent stages, E1 and E2, each with the same transition rate constant. Newly infected individuals are in class E1, pass to class E2 with a 'partial incubation' process having rate constant σ, and then pass on to class I with a second partial incubation process also having rate constant σ. Prepare a compartment diagram for this SEEIR model like that of Figure 5.

Exercise 14. (Calculus Level 2, Programming Level 0)
Prepare a compartment diagram like that of Figure 5 for a model in which a fraction p of infectious individuals are asymptomatic (class A) and the remainder are symptomatic (class I). Assume that asymptomatic individuals do not subsequently become symptomatic. In completing the diagram, keep in mind that both classes in this SEAIR model are infectious.

6 Modeling

In addition to variables and equations, mathematical models contain *parameters*. An understanding of parameters is critically important in mathematical modeling.

Definition 5. Parameters are quantities that function either as variables or constants, depending on the context.

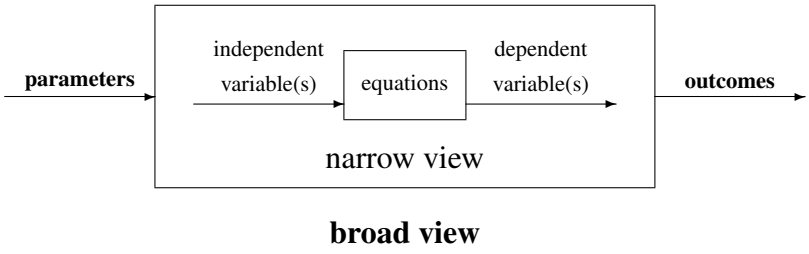

Fig. 6 Narrow and broad views of mathematical models.

The critical importance of parameters is illustrated in the schematic diagram of modeling shown in Figure 6. In developing the model, we had time as the independent variable and dependent variables S, E, I, and R. In this *narrow* view, the parameters are constants. This is the realm of mathematics, be it analytical calculation (algebra and/or calculus) or numerical simulation. We can do analysis without parameter values, but we treat them the same way we treat the constant π; that is, as numbers denoted by symbols rather than numerals. Numerical simulation requires specific values for the parameters to produce numerical approximations for the rates of change at specific times.

Most important modeling work takes place in the *broad* view. Here is where we ask questions that we hope will be useful in interpreting the scenario that inspired the model rather than simply calculating results. In the broad view, we can study the effects of changes in the parameter values, asking questions such as "If two diseases have different incubation periods, how does that difference affect the outcomes of the model?" The outcomes can be whatever we are most interested in: for example, the maximum size of the infectious class, the day on which that occurs, and/or the total number of people who get the disease during the outbreak. Such information could be used to estimate the danger of running out of hospital space and the total number of deaths, for example.

The function concept is helpful in understanding how mathematical models work. In the narrow view, the class counts are functions of the number of days since the start of the outbreak. In the broad view, the outcomes are functions of the parameter values. Both the narrow view and broad view functions differ in an important way from the functions you are used to seeing in math classes. Functions in math classes are nearly always defined by explicit formulas, such as the formula for the incubation rate (9). In mathematical modeling, functions are defined in much more subtle ways; for example, as the solution of a mathematical problem rather than as a mathematical formula. Conceptually, however, the function idea is the same regardless of whether the function is computed from a formula or as the result of a numerical procedure. It is best to think of functions in terms of their graphs rather than their formulas. Sections 7 and 8 develop tools for working with the functions in models. In this section, we focus on the specification of parameters.

6.1 Identifying Parameter Values

Initial condition parameters are determined by the specific scenario we have in mind. In practice, we'll assume that the initial fractions of removed, latent, and infectious individuals are known, with everyone else susceptible. How we do this depends on the starting point for the scenario. For the initial introduction of a disease to a community, we assume that nobody is immune or infectious and that just a small fraction of people are latent. Other situations will be discussed later.

The three process parameters represent the quantitative characteristics of the particular disease being modeled. Often we have a specific disease in mind, so it is important to be able to use measurable properties to determine the values.

Dimensional analysis provides clues for the meaning of parameters. Things that can be compared in an equation or added have to have the same dimension, and the dimension of a products is the product of the dimensions. The terms dI/dt and γI must have the same dimension of people/time, so γ has dimension 1/time. What is not clear without mathematical development is how to know what time γ is the reciprocal of. The theorem given here is proven in Challenge Problem 2.

Theorem 1. *A transition rate parameter is the reciprocal of the average amount of time required for the transition process.*

While simple, this rule is actually very subtle. In order to derive it, we have to solve a calculus problem; hence, we defer this to the exercises.

Let t_L be the mean incubation period and t_I the mean infectious duration, which are generally known to a reasonable approximation. Then the transition rate parameter rule gives us

$$\eta = \frac{1}{t_L}, \qquad \gamma = \frac{1}{t_I}. \tag{17}$$

Note that the values of these parameters do not depend on whether we use population counts or population fractions for the state variables.

The transmission rate parameter β poses a special problem. It is not actually a fundamental disease property because its numerical value depends on how we define population size. It does not have the dimension of 1/time, so it can't be the reciprocal of a measured time. Fortunately, there is a parameter that represents the infectiousness of a disease.

6.2 The Basic Reproductive Number

The *basic reproductive number*, which is given the symbol $\mathscr{R}_0$ and read as 'R-nought', is the fundamental measure of the infectiousness of a disease.

Definition 6. The **basic reproductive number** is the average number of secondary infections brought about by one infectious person in a population that is wholly susceptible.

This definition of $\mathscr{R}_0$ sounds complicated, but it is much simpler if we break it down into parts. The transmission rate formula (11) tells us that the average number of secondary infections *per day* in a population of any composition is βSI. If that population is wholly susceptible, then we get an average of βNI secondary infections per day. That is the number produced by the whole infectious class; with $I = 1$, we see that 'the average number of secondary infections *per day* brought about by one infectious person in a population that is wholly susceptible' is βN. To get the basic reproductive number, we just need to take into account that one infectious person has, on the average, t_I days in which to produce secondary infections. Total is rate times time, so the basic reproductive number is

$$\mathscr{R}_0 = \beta N t_I. \tag{18}$$

While we have calculated the basic reproductive number in terms of β, the actual use of this formula is usually to determine β from $\mathscr{R}_0$. Of course this means that we need to be able to determine $\mathscr{R}_0$ by some other means. You can look up values for well known diseases. The issue of how to estimate $\mathscr{R}_0$ for a novel disease is critically important to accurate modeling results; this will be addressed in Section 7.

Note that the value of $\mathscr{R}_0$ is independent of the units used for population class sizes. If we change N, it is the value of β that makes a corresponding change. Given

that we have chosen $N = 1$ by design, we can rearrange (18) and replace t_I with $1/\gamma$ to obtain

$$\beta = \gamma \mathscr{R}_0. \tag{19}$$

We can therefore specify a particular disease using t_L, t_I, and $\mathscr{R}_0$ as the three fundamental disease parameters and use (17) and (19) to calculate the parameters that appear in the model.

Before we move on, this is a good time to think about what the basic reproductive number tells us. Suppose the first generation consists of 10 infectious individuals. If $\mathscr{R}_0 = 3$, then each infectious person will generate an average of three new infections, for a total of 30 in the second generation. The epidemic will grow explosively as long as the population remains largely susceptible. Only when most of the population has been infected will we stop seeing more infections in the next generation than the previous one. In the end, nearly everyone will have got the disease.

Instead, if $\mathscr{R}_0 = 1.1$, there will be an average of 11 in the second generation. This is enough to keep the outbreak growing for a little while, but at a much slower rate than if $\mathscr{R}_0 = 3$, and a much smaller decrease in the susceptible population will be enough to stop the disease. Continuing with the comparison, if $\mathscr{R}_0 = 0.9$ then the second generation will average 9 individuals. We see that the disease is unable to get a foothold. Thus, $\mathscr{R}_0 = 1$ is a critical value—a disease can only cause an epidemic outbreak if $\mathscr{R}_0 > 1$.

We can now appreciate why COVID-19 is such a serious problem. The most common infectious diseases today, the common cold and influenza, have $\mathscr{R}_0$ values on the order of 1.5 to 3. It is not uncommon for a person to have known exposures to someone with the flu and not get the disease. In contrast, the standard 20th century childhood diseases of measles, chicken pox, and mumps have $\mathscr{R}_0$ values of 10 or more. Before the development of the vaccines for these diseases, virtually everyone who was exposed caught them.[14] The basic reproductive number for the original strain of COVID-19 has been reported to be in the range from 2.5 to 8. Our best estimate is $\mathscr{R}_0 = 5.7$ [14], which is a very large value.[15] When you run simulations, you will see that in a society that completely ignores the threat, almost the entire population will get the disease in less than 2 months.

Section 6 Exercises

Exercise 15. (Calculus Level 0, Programming Level 0)
Estimates of the basic reproductive number and durations for incubation and infec-

[14] Curiously, neither of the chapter authors, both growing up before there were vaccines for these diseases, was ever diagnosed with chicken pox. Nevertheless, one of the authors is now known through detection of antibodies to have had the disease, and the other almost certainly had the disease as well because of repeated exposures.

[15] As of July 2021, the delta variant is just becoming dominant, with a basic reproductive number probably between 8 and 10. Given this value, virtually everyone who is not vaccinated will get the disease.

tiousness can be found online for many illnesses. Use published information about H1N1 flu [3] to estimate reasonable parameter values for η, γ, and β. Be sure to justify all assumptions and cite all references.

Exercise 16. (Calculus Level 0, Programming Level 0)
Repeat Exercise (15) using the infectious disease of your choice. Justify all assumptions and cite all references.

Section 6 Challenge Problem

Challenge Problem 2. (Calculus Level 2, Programming Level 0)
Assume a population of newly infectious people and let $I(t)$ be the fraction of this cohort that is still infectious at time t, so that

$$\frac{dI}{dt} = -\gamma I, \quad I(0) = 1.$$

Use your knowledge of derivative formulas to identify the function $I(t)$ that satisfies these two requirements. Rewrite the result by solving it for t; this gives you the time at which the cohort has been reduced to a particular size. Now use calculus to determine the average value of t over the interval $0 \leq I \leq 1$. Explain why this proves Theorem 1.

7 Model Analysis

It is helpful to frame the model analysis by using the idea of a mathematical model as a function that determines outcomes in terms of the parameters. The details are shown in Figure 7 as a schematic diagram.

Fig. 7 Schematic diagram of the SEIR model as a function.

There are six input parameters, the three that define the disease properties and three that define the starting point of the scenario. There are a number of possible outcomes we could be interested in. We focus on five of these:

1. The early-stage exponential growth coefficient, λ (this will be defined shortly), which tells us how rapidly the epidemic grows at the beginning;
2. The maximum infectious class size, I_{max}, which tells us how much impact the epidemic will have at the worst point;
3. The time at which the maximum infectious class size occurs, t_{max}, which tells us how much time there is to prepare for the peak;
4. The ending susceptible population, denoted as s_∞, which tells us how much of the population does not contract the illness and will be at risk in a subsequent scenario;
5. The total population infected during the scenario, $\Delta S = s_0 - s_\infty$. If we know what fraction of infected people die, we can use ΔS to calculate the total number of deaths in a given initial population.

Our outcomes will need to be determined by a variety of methods. We can use analytical methods (exact calculations using algebra or calculus) to compute λ and to derive an algebraic equation for s_∞. That equation cannot be solved using analytical methods, but it can be solved with numerical methods (approximate calculations using a computer program). Most epidemic model outcomes, such as I_{max} and t_{max} in the SEIR model, can only be determined by a fully numerical method. In this section, we consider λ and s_∞; the following section discusses numerical simulation of the full model.

7.1 Early-Phase Exponential Growth

Analytical methods and numerical methods are complementary in many ways. One of these is that the behavior illustrated by numerical simulations can yield conjectures that can be subsequently confirmed by analysis. Figure 8 shows the results of a simulation using an initial condition of no infectious or removed individuals and only one latent individual per 10K population. The plot on the left shows the typical pattern of an epidemic outbreak. It takes a while to get started, but then the infection grows rapidly. Both the latent and infectious classes reach a peak and then drop off to 0, with the latent peak occurring earlier in time than the infectious peak. The plot on the right shows some very important detail that can only be seen on a logarithm plot. There is a very fast initial phase during which the latent population is decreasing. This is because we started without any infectious individuals, so new transmissions had to wait until the first batch of latent individuals became infectious. Then there is a significant period of time during which the graphs of $\ln(E)$ and $\ln(I)$ are linear *and parallel*. Only as the latent population comes close to its peak do the graphs of $\ln(E)$ and $\ln(I)$ begin to curve downward. These same features appear with any realistic choices for the disease parameters, as long as the initial fractions of the exposed and infectious classes are small. From this graph, we can make the following conjecture:

- After a short initial adjustment phase, there is a period in which the logarithms of the infected classes are linear with a common slope λ.

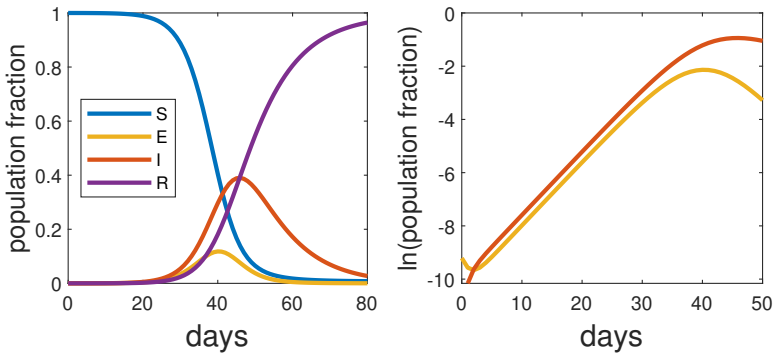

Fig. 8 Simulation results for the continuous model with $R_0 = 5$, $t_L = 2$, $t_I = 10$, $e_0 = 0.0001$, $i_0 = r_0 = 0$

Mathematical exploration of this insight leads to the following result (see Challenge Problem 3):

Theorem 2. *Suppose the initial populations of the infected classes are small compared to that of the susceptible class. Then the SEIR model shows an extended exponential growth phase with*

$$I \approx I_0 e^{\lambda t}, \quad E \approx \rho I_0 e^{\lambda t}, \quad S \approx s_0 \approx 1 - r_0, \tag{20}$$

where λ is the positive solution of the equation

$$(\lambda + \eta)(\lambda + \gamma) = \eta \gamma s_0 \mathscr{R}_0, \tag{21}$$

and

$$\rho = \frac{\lambda + \gamma}{\eta}. \tag{22}$$

Theorem 2 has two very important consequences. First, it gives us a way to estimate $\mathscr{R}_0$ from early data on the infectious class population. From data for $\ln(I)$, we can estimate the value of λ and then use (21) to estimate $\mathscr{R}_0$. This is the best way to estimate $\mathscr{R}_0$ for a novel disease, like COVID-19. In late March of 2020, the 'accepted' value of $\mathscr{R}_0$ for COVID-19, obtained by statistical analysis of known transmissions, was 2.6. One of us (GL) used (21) to obtain an estimate of 5.0 at that time. Subsequently, our current best guess of 5.7, obtained using a method similar to ours, was published in July 2020 [14].

The other important consequence of Theorem 2 is that it allows us to prescribe scenarios using only two initial conditions rather than three. For any scenario that starts with a small infected population, we can choose i_0 and then use $e_0 = \rho i_0$, where ρ is given by (22).

7.2 The End State

Dynamical systems often progress toward a fixed end state. These must be states for which all of the rates of change are 0. For endemic models, there are usually only one or two such states and there are straightforward methods to determine which is the end state. The situation is much more complicated for epidemic models. To begin, note that a fixed end state must have no further changes in R. Since $dR/dt = \gamma I$, we can only have a fixed value of R if $I = 0$. From here, we can similarly conclude that a fixed value $I = 0$ also requires $E = 0$. The chain stops here, however. There is no particular reason why a fixed end state should have $S = 0$ or any particular value of R. Based on the differential equation model (12–16), any state with values

$$S = s_\infty, \quad E = 0, \quad I = 0, \quad R = r_\infty = 1 - s_\infty \tag{23}$$

could serve as the end state.

In the case of the SEIR epidemic model and a few other simple ones, the end state can be found using calculus. The idea is that the relationship between the variables S and R is determined by the differential equations for those two variables. If we assume that R is a function of S, rather than being an independent function of t, we can use the chain rule to obtain an expression for dR/dS. If that expression depends only on the variable S, as is the case here, then we can use integration techniques to find the one-parameter family of anti-derivatives for dR/dS. Only one of these anti-derivatives also satisfies the initial conditions for R and S. The result is as follows:

Theorem 3. *The initial and final values of the susceptible population are related by the equation*

$$\ln s_0 - \ln s_\infty = \mathscr{R}_0(1 - r_0 - s_\infty). \tag{24}$$

Equation (24) is an analytical result that connects the initial values r_0 and s_0, the basic reproductive number $\mathscr{R}_0$, and the final value s_∞. It cannot be used to immediately calculate s_∞ for any set of input parameters because there is no way to solve the algebraic equation for s_∞. However, we can obtain several useful conclusions from the theorem:

1. The final state s_∞ depends only on the basic reproductive number and the initial conditions; it is unaffected by the values used for the rate constants η and γ. If these constants are relatively small for one scenario, then it will just take longer to reach the same final state.

2. The final state cannot have $s_\infty = 0$ because there are no values one can choose for the other parameters to satisfy (24).
3. If we want to see how s_∞ depends on the infectiousness of the disease for a given initial scenario, we can use (24) to calculate $\mathscr{R}_0$ values from given values of s_∞ and then plot a graph of s_∞ vs $\mathscr{R}_0$, without ever solving the equation for s_∞.

If we want to be able to calculate s_∞ rather than approximating it from a graph, we need a numerical method, since (24) can't be solved for it using algebra. The equation can be manipulated in various ways to put it in the form $F(s_\infty) = 0$. There are a variety of well-documented numerical methods for solving such equations. These methods are much harder to use in practice than they are in theory because of various complicating factors. Scientific computing software, such as Matlab, includes built-in functions for this task. These have been written by numerical analysis experts, and you can use these if you just want an answer. If you want better mathematical understanding, you have to write your own programs to test specific methods. We include this as a possible project for the more mathematically experienced.

Section 7 Exercises

Exercise 17. (Calculus Level 0, Programming Level 0) Find estimates for the incubation period, the infectious period, and the basic reproductive number for an infectious disease of your choice. Use this information to calculate η, γ, λ, and ρ for that disease. Also estimate s_∞ assuming $r_0 = 0$ and $s_0 \approx 1$, either by using a numerical solver with (24) or by using a plot of s_∞ vs $\mathscr{R}_0$ (see consequence 3 of Theorem 3).

Exercise 18. (Calculus Level 1, Programming Level 0)
As described in the opening paragraphs of Section 7, the graph on the left of Figure 8 shows both the latent and infectious classes reach a peak before dropping off to 0. Note that in addition to occurring earlier in time, the latent peak is lower and approaches zero more quickly than the infectious peak. Explain why this is the case for the given parameters. Would it be possible for the latent peak to be larger than the infectious peak? Would it be possible for the latent curve to be greater than the infectious curve during certain periods of time during the epidemic? If so, for what parameters might this happen? Explain.

Exercise 19. (Calculus Level 1, Programming Level 0)
Explain what the right panel of Figure 8 suggests about the nature of the epidemiological process in the stage where the graphs of $\ln(E)$ and $\ln(I)$ are parallel. It may help to reference material in Section 1. Your explanation should also relate the plot to the corresponding plot in the left panel.

Section 7 Challenge Problems

Challenge Problem 3. (Calculus Level 2, Programming Level 0)
Assume that $S = s_0$, $R = r_0$, $\ln I = \ln I_0 + \lambda t$, and $\ln E = \ln I + \ln \rho$ for some unknown values I_0 and ρ (note that these assumptions are justified when e_0 and i_0 are small by Figure 8). Derive the results of Theorem 2 by substituting these assumptions into the model (12–15) .

Challenge Problem 4. (Calculus Level 2, Programming Level 0)
Assume that $\mathscr{R}_0$, η, γ, i_0, and r_0 are given, with i_0 small. Use the same method as in Challenge Problem 3 to derive a quadratic equation whose solution determines ρ. Note that this formula can be used to select an appropriate e_0 for a scenario that starts shortly after the beginning of an outbreak.

Challenge Problem 5. (Calculus Level 2, Programming Level 0)
Assume R is a function of $S(t)$. Use this assumption and the model (12–15) to derive a formula for dR/dS. Then use calculus and algebra to derive the result of Theorem 3.

Section 7 Project

Research Project 6. (Calculus Level 2, Programming Level 2)
Develop an algorithm for solving (24) for s_∞. If you use a method, such as Newton's method, that requires a starting guess, your algorithm should include steps that compute a good starting guess from the parameters s_0, r_0, and $\mathscr{R}_0$. Your algorithm should work for reasonable values of the parameters; for example, $1 < \mathscr{R}_0 \leq 10$.

8 Simulations

In Section 7, we identified five important outcomes for an epidemic model. Even when some of these can be determined analytically, there are always some that can only be determined by running a numerical simulation. We also need a numerical simulation if we want to see the full course of the epidemic rather than just seeing key outcomes.

There are two different aspects of simulation that we need to study: the numerical aspect and the modeling aspect. The distinction is best made in terms of Figure 6. The narrow view is the realm of numerical analysis; we need to work out methods to determine the time histories of the state variables S, E, I, and R, with a given

set of parameters. The broad view is the realm of modeling; we need to work out strategies for getting useful information from data for the time histories.

8.1 Numerical Simulation of Continuous Dynamical Systems

Continuous dynamical systems pose a problem for simulation because computers can only do algebra, not calculus. This means that the only way to simulate a continuous dynamical system is to approximate it by a discrete dynamical system. Consider the model $dy/dt = f(y) = by(1-y)$, which represents an SI model, as an example. The obvious way to approximate the differential equation is to replace the derivative with the difference quotient $\Delta y/\Delta t$. Then we can define y_n to be the approximate value of y on day n. With f evaluated using the population on day n, we have

$$y_{n+1} = y_n + f(y_n)\Delta t. \tag{25}$$

(Note that $\Delta t = 1$ in our model.) This straightforward method is called *Euler's method*.[16] While this is the most intuitive way to approximate a differential equation numerically, there are much better methods. In most cases, including the models that arise in epidemiology, the method of choice is a more complicated method called the Runge-Kutta[17] method of order 4, sometimes referred to as 'RK4'. The basic idea of the method is that we want to retain the simple structure of Euler's method,

$$y_{n+1} = y_n + m\Delta t,$$

but with a more sophisticated estimate of the slope m. The RK4 method calculates three other slope estimates in addition to $f(y_n)$ and takes a weighted average. A more complete discussion can be found in most differential equations texts, and the formulas appear in the program *seir.m* in the Appendix.

In general, one can improve accuracy by using time intervals smaller than 1 day, but this is unnecessary when using the RK4 method with an epidemiology model because the time scales for the processes are on the order of several days.

8.2 Implementation of Numerical Simulations

We present here a brief description of the Matlab program suite that we use to implement the numerical simulation of the SEIR model (the programs themselves appear in the Appendix). We assume that students have Matlab installed and know how to access the software. or else know how to run Matlab programs using the online

[16] Pronounced OY-ler.

[17] Pronounced RUN-guh KUT-tah.

version of Octave [12]. The reader should review the description of the individual-based model programs in Subsection 2.2.

One way to make an accessible program suite is to have it consist of a function file that does computations for one scenario and a set of driver scripts that use the computational function to do experiments. The function file *seir.m* contains the function *seir*, which requires input values for the basic reproductive number, the mean incubation time, the mean recovery time, initial conditions, and data used to implement a situation-based ending condition. The outputs of *seir* are vectors that give the values of the four population classes at each time along with the value of s_∞ calculated by applying Matlab's built-in equation solver to solve (24).

Most of the coding in *seir.m* can be left as is when modifying the program to simulate other models. Additional function arguments and outputs may be needed, and there will likely be small changes to the *INITIALIZATION* section, but most of the changes will appear in the *FUNCTION FOR THE DIFFERENTIAL EQUATION* section. The comments in this section and the syntax of the statements in it should suffice for a novice programmer to be able to make modifications as needed.

There are three driver programs, each designed for a particular type of experiment:

- *SEIR_onesim.m* runs one simulation, leading to a plot like that in the left panel of Figure 8. The user needs to modify the *SCENARIO DATA* section for each run.
- *SEIR_comparison.m* runs a comparison of scenarios having one parameter that takes a variety of values. There is a section for default scenario data that includes all the input parameters. Values for the parameter selected as the independent variable are specified as *xvals* in the *INDEPENDENT VARIABLE DATA* section. The first line in the computation loop assigns values from *xvals* to the desired parameter name. This uses a different value for that parameter on each run through the loop, while the other parameters keep their default values.
- *SEIR_paramstudy.m* produces plots of key outcomes as a function of one of the parameters. The parameter values are specified in almost the same manner as for *SEIR_comparison*.

Section 8 Exercises

Exercise 20. (Calculus Level 1, Programming Level 1)
Our best guesses for the original strain of COVID-19 are a basic reproductive number of 5.7, an incubation period average of 5 days, and an infectious period average of 10 days. Assume an initial population that is entirely susceptible except for ten latent individuals per 100K. Run *SEIR_onesim.m* and describe what would have happened in a community that made no behavioral or public health adjustments.

Exercise 21. (Calculus Level 1, Programming Level 1)
Repeat Exercise 20 using the disease you characterized in Exercise 17. Make sure

that the final susceptible fraction obtained by the computer simulation matches the one you estimated in that exercise.

Exercise 22. (Calculus Level 1, Programming Level 1)
The Incan Empire had a population of over one million when it was conquered by 168 Spanish Conquistadores in 1525. The Spanish had gunpowder weapons and horses, but these advantages would not have been sufficient to defeat the huge Incan army. (It took about 2 minutes to reload a single-shot arquebus.) They also benefitted by joining forces with peoples subjugated by the Incas, but that would not have happened on its own. Historians William H. McNeill and Jared Diamond have argued that the key factor in the Incan defeat was the European diseases the Spanish brought with them [11, 4]. To test this theory, set the basic reproductive number at 5, the incubation period at 12 days, and the infectious duration at 20 days, values that roughly match smallpox. Describe the effect introduction of smallpox into Incan civilization would have had, even without considering the death toll of the disease.

Exercise 23. (Calculus Level 1, Programming Level 1)
For a more complete look at the effect of the basic reproductive number on epidemic progression, run *SEIR_comparison.m* using *R0* values 5, 3, 2, 1.5, and 1.25, with incubation period 5 days, infectious duration 10 days, and initial infectious fraction 0.001.

1. Discuss the graphs, explaining why the effects of *R0* are what you see.
2. Suppose a disease with *R0*=5 is combated with social distancing and measures to decrease transmission probability for each contact. What effect do you expect these social policies to have and why?

Exercise 24. (Calculus Level 1, Programming Level 1)
Use *SEIR_paramstudy.m* to study the effect of the basic reproductive number on epidemic outcomes. Use *R0* values from 0 to 6 and the original default values for the other parameters. Describe and explain the results, paying particular attention to the behavior near *R0*=1.

Exercise 25. (Calculus Level 1, Programming Level 1)
Use *SEIR_paramstudy.m* to do a more thorough study of the effect of the disease duration on epidemic outcomes. Use *R0*=2.5 and *tI* values from 4 to 12. Describe and explain the results. Pay particular attention to the axis limits.

Section 8 Projects

Research Project 7. (Calculus Level 1, Programming Level 1)
Write an analysis of herd immunity. You can find basic information about

this concept on the web, but you will need to justify all of your claims with simulation results rather than quotations from other sources. You will have to think carefully about what outcome is the most important to report. You should also address the question of why herd immunity does not work in an epidemic scenario for a disease with a large value of $\mathscr{R}_0$ (say 5.0) that starts with nearly everyone susceptible.

Research Project 8. (Calculus Level 1, Programming Level 2)
How much of a problem is the assumption that transition rates are proportional to the class size? At issue is the association of this assumption with the exponential distribution. Look up the characteristics of this distribution. A more realistic choice is the Erlang E_2 distribution. An Erlang E_2-distributed process is equivalent to a sequence of two exponentially-distributed stages. This means that we can get Erlang-distributed incubation times by replacing the standard SEIR model with the SEEIR model from Exercise 13. To study this new model, you will need to modify the SEIR programs for SEEIR and compare the results. Be careful about how you calculate σ; keep in mind that t_L is the average total time spent in the E classes, not the average time spent in each of the classes. Under what circumstances are we making a significant error by using the SEIR model in place of the more accurate SEEIR model?

Research Project 9. (Calculus Level 1, Programming Level 2)
Research the impact of isolation of symptomatic infectious patients on the course of an epidemic. This will require you to change the model from SEIR to SEPUR, where the original infectious class has been divided into a sequence of a presymptomatic class and an unisolated symptomatic class. You can assume that some fraction q of individuals who transition out of the presymptomatic class move directly into the removed class through isolation, while the remainder move into the unisolated class and become removed in the usual way. Note that you have to be careful with your formula for the transmission process, since the new model has two infectious classes. Exercise 12 can serve as a good starting point.

Research Project 10. (Calculus Level 1, Programming Level 2)
Build and study a model that incorporates social distancing and/or mask us-

age to decrease transmission. Consider making the model more realistic by dividing the population into a compliant subgroup and a noncompliant subgroup, but keep in mind that individuals in the two groups still interact with each other.

Research Project 11. (Calculus Level 1, Programming Level 2)
How does the development of a vaccine change the course of an epidemic? You will need a new model that accounts for several features of vaccine implementation: (1) a new vaccine can only be distributed gradually as doses are produced and distributed, (2) not everyone is willing to take a vaccine, and (3) not everyone who is vaccinated develops immunity.

Appendix: Programs

hpsr.m

```
function [H,P,S,R]=hpsr(b,N,P0,S0)
%
% function [H,P,S,R]=hpsr(b,N,P0,S0)
%
%   runs a simulation of an agent-based model
%
%   H: healthy
%   P: presymptomatic
%   S: sick
%   R: recovered
%
%   b is the transmission probability
%   N is the total population
%   P0 is the initial presymptomatic population
%   S0 is the initial sick population
%
% by Glenn Ledder
% written 2020/11/29
% revised 2021/10/20
%
% direct comments to gledder@unl.edu

%% DATA
```

```
% suggested default values
% b = 1;
% N = 1000;
% P0 = 20;

%% INITIALIZATION

% limit simulation duration
maxdays = 100;

% set data structures using initial conditions
H = (N-P0-S0)*ones(1,maxdays+1);
P = P0*ones(1,maxdays+1);
S = S0*ones(1,maxdays+1);
R = zeros(1,maxdays+1);

%% COMPUTATION

for n=1:maxdays

    % calculate probability of an infected partner for a healthy person
    p = (P(n)+S(n))/(N-1);

    % get a 'partner' for each healthy person
    partners = rand(H(n),1);

    % count H's with infectious partners
    atrisk = length(find(partners<p));

    % get random numbers for transmission success
    exposure = rand(atrisk,1);

    % count atrisks with successful transmission
    infected = length(find(exposure<b));

    % calculate new class counts
    H(n+1) = H(n)-infected;
    P(n+1) = infected;
    S(n+1) = P(n);
    R(n+1) = R(n)+S(n);

    % check if done
    if P(n+1)+S(n+1)==0
        % delete unneeded rows
        H = H(1:(n+1));
```

```
        P = P(1:(n+1));
        S = S(1:(n+1));
        R = R(1:(n+1));
        % exit for loop
        break;
    end %if

end %for

%% END

end
```

HPSR_onesim.m

```
%% HPSR_onesim

% Plots the results of an HPSR agent-based model simulation

% Prints results and outcomes:
%   results is a matrix of columns for time,H,P,S,R
%   maxI is the maximum number infected
%   finalH is the ending size of H

% User specifies values for 4 parameters:
%   b is the transmission probability
%   N is the total population
%   P0 is the initial presymptomatic population
%   S0 is the original sick population

% If using Octave, add the additional option pair "'MarkerSize',2.5"
%   (without the double quotes) to the three plot statements)

% Uses hpsr.m, 2021/10/20 version

%- function [H,P,S,R]=hpsr(b,N,P0,S0)
%-
%-   runs a simulation of the HPSR agent-based disease model
%-
%-   H: healthy
%-   P: presymptomatic
%-   S: sick
%-   R: recovered
```

```
%-
%-   b is the transmission probability
%-   N is the total population
%-   P0 is the initial presymptomatic population
%-   S0 is the initial sick population

% by Glenn Ledder
% written 2020/11/29
% revised 2021/10/26

% direct comments to gledder@unl.edu

%% SCENARIO DATA

b = 5/6;
N = 1000;
P0 = 20;
S0 = 0;

%% INITIALIZATION

% set up for first figure

figure(1)
clf
darkgreen = [0 0.6 0];
hold on
box on

%% COMPUTATION

% collect simulation data
[H,P,S,R] = hpsr(b,N,P0,S0);

% record simulation duration [H(1) is for day 0]
days = length(H)-1;

%% OUTPUT

% get list of times for plot
times = 0:days;

% If using Octave, add the additional option pair "'MarkerSize',2.5"
%   (without the double quotes) to the three plot statements
```

```
% plot healthy, infected, recovered
plot(times,H,'d','Color',darkgreen,'MarkerFaceColor',darkgreen)
plot(times,P+S,'rs','MarkerFaceColor','r')
plot(times,R,'b^','MarkerFaceColor','b')
xlim([0,days])

% label axes
%   use 'FontSize',18 in Octave
xlabel('days','FontSize',14)
ylabel('populations','FontSize',14)

legend('healthy','infected','recovered','Location','East')

% display results as a matrix with columns for t, H, P, S, R
results = [times',H',P',S',R']

% report maximum simultaneously infected
maxI = max(P+S)

% report final healthy population
finalH = H(end)

%% FIGURE 2

% uncomment the lines from 101 to get a figure of log(P+S)
%   add ",'MarkerSize',2.5" to the end of this plot command in Octave
%   use 'FontSize',18 in Octave

% figure(2)
% clf
% hold on
% box on
%
% plot(times,log(P+S),'rs','MarkerFaceColor','r')
%
% xlabel('days','FontSize',14)
% ylabel('ln(infected)','FontSize',14)
```

HPSR_avg.m

```
%% HPSR_avg

% Computes results for an HPSR agent-based model simulation
```

```
% Displays histograms and prints means and standard deviations of outcomes
%   maxI_pct is the maximum number infected by percentage
%   finalH_pct is the ending size of H by percentage

% User specifies values for 5 parameters:
%   b is the transmission probability
%   N is the total population
%   P0 is the initial presymptomatic population
%   S0 is the original sick population
%   numruns is the number of simulation runs in the experiment

% SEE SPECIAL INSTRUCTIONS (below) FOR HISTOGRAMS IF USING OCTAVE.

% Uses hpsr.m, 2021/10/20 version

%- function [H,P,S,R]=hpsr(b,N,P0,S0)
%-
%-   runs a simulation of the HPSR agent-based disease model
%-
%-   H: healthy
%-   P: presymptomatic
%-   S: sick
%-   R: recovered
%-
%-   b is the transmission probability
%-   N is the total population
%-   P0 is the initial presymptomatic population
%-   S0 is the initial sick population

% by Glenn Ledder
% written 2020/12/30
% revised 2021/10/26

% direct comments to gledder@unl.edu

%% SCENARIO DATA

b = 1;
N = 1000;
P0 = 10;
S0 = 0;
numruns = 1000;

%% INITIALIZATION
```

```
% set up for figure

clf
for k=1:2
    subplot(1,2,k)
    hold on
end

% create row vectors for output data

maxI = zeros(1,numruns);
finalH = zeros(1,numruns);

%% COMPUTATION

for j=1:numruns
    % run simulation
    [H,P,S,~] = hpsr(b,N,P0,S0);
    % save results
    maxI(j) = max(P+S);
    finalH(j) = H(end);
end

% convert to percentages and sort
maxI_pct = 100*sort(maxI)/N;
finalH_pct = 100*sort(finalH)/N;

%% OUTPUT

subplot(1,2,1)
% For Octave, replace the histogram statement with the following:
% hist(maxI_pct,14,1,'FaceColor','y')
histogram(maxI_pct,'Normalization','probability','FaceColor','y','FaceAlpha',1
xlabel('max I (pct)','FontSize',11)
ylabel('probability','FontSize',11)

subplot(1,2,2)
% For Octave, replace the histogram statement with the following:
% hist(finalH_pct,14,1,'FaceColor','y')
histogram(finalH_pct,'Normalization','probability','FaceColor','y','FaceAlpha'
xlabel('final H (pct)','FontSize',11)
ylabel('probability','FontSize',11)

% report means and standard deviations
```

```
maxI_pct_mean = mean(maxI_pct)
maxI_pct_std = std(maxI_pct)
finalH_pct_mean = mean(finalH_pct)
finalH_pct_std = std(finalH_pct)
```

seir.m

```
function [S,E,I,R,sinfty]=seir(R0,tL,tI,e0,i0,r0,target,maxdays)
%
% function [S,E,I,R,sinfty]=seir(R0,tL,tI,e0,i0,r0,target,maxdays)
%
% runs a simulation of an SEIR model
%
% S: susceptible
% E: 'exposed' ('latent' is the more accurate term)
% I: infectious
% R: removed
%
% R0 is the basic reproductive number
% tL is the mean incubation time
% tI is the mean recovery time
% e0 is the initial latent fraction
% i0 is the initial infectious fraction
% r0 is the initial immune fraction
% target is the infected fraction used as an end condition
% maxdays is the maximum simulation duration
%
% by Glenn Ledder
% written 2020/11/27
%
% direct comments to gledder@unl.edu

%% DATA

% suggested default values
% R0 = 5;
% tL = 2;
% tI = 10;
% e0 = 0.0001;
% i0 = 0;
% r0 = 0;
% target = 0.001;
% maxdays = 1000;
```

```
%% INITIALIZATION

% calculate parameters

eta = 1/tL;
gamma = 1/tI;
beta = R0*gamma;
s0 = 1-e0-i0-r0;
k = 1-r0-log(s0)/R0;

% set up results data structure with Y=[S,E,I,R]

results = zeros(maxdays+1,4);
Y = [s0,e0,i0,r0];
results(1,:) = Y;

y = Y';
oldx = e0+i0;

%% FUNCTION FOR Sinfty

fnc = @(s,R0,k) s-exp(R0*(s-k));    % parameterized function
fofs = @(s) fnc(s,R0,k);            % function of s alone

%% COMPUTATION

for t=1:maxdays
    % y is a column vector, Y^T
    y = rk4(1,y);
    Y = y';
    results(t+1,:) = Y;
    if Y(2)+Y(3)>min(target,oldx)
        oldx = Y(2)+Y(3);
    else
        results = results(1:(t+1),:);
        break;
    end
end

S = results(:,1);
E = results(:,2);
I = results(:,3);
R = results(:,4);
```

```
sinfty = fzero(fofs,[0,1]);

%% FUNCTION FOR rk4

    function y=rk4(dt,y0)
        % y0 is a column vector of initial conditions at t
        % y is a column vector of values at t+dt
        k1 = yprime(y0);
        k2 = yprime(y0+0.5*dt*k1);
        k3 = yprime(y0+0.5*dt*k2);
        k4 = yprime(y0+dt*k3);
        y = y0+dt*(k1+2*k2+2*k3+k4)/6;
    end

%% FUNCTION FOR THE DIFFERENTIAL EQUATION

    function yp=yprime(y)
    % split out components
        S = y(1);
        E = y(2);
        I = y(3);
    % compute derivatives
        Sp = -beta*S*I;
        Ep = -Sp-eta*E;
        Ip = eta*E-gamma*I;
        Rp = gamma*I;
    % assemble derivative
        yp = [Sp;Ep;Ip;Rp];
    end

%% END

end
```

SEIR_onesim.m

```
%% SEIR\_onesim

% Plots a comparison of population classes

% Prints results and outcomes (populations are fractions):
%   results is a matrix of columns for time,S,E,I,R,Delta S
%   maxI is the maximum size of class I
```

```
%   maxday is the day on which maxI occurs
%   finalS is the ending size of S

% Uses seir.m, 2020/11/27 version

% User specifies values for 6 parameters:
%   R0 is the basic reproductive number
%   tL is the mean incubation time
%   tI is the mean recovery time
%   e0 is the initial latent fraction
%   i0 is the initial infectious fraction
%   r0 is the initial immune fraction
%   target is the infected fraction used as an end condition
%   maxdays is the maximum simulation duration

% Change axis label font sizes for Octave

% by Glenn Ledder
% written 2020/11/27
% revised 2021/08/01

% direct comments to gledder@unl.edu

%% SCENARIO DATA

R0 = 5;
tL = 2;
tI = 10;
e0 = .0001;
i0 = 0;
r0 = 0;

%% COMMON DATA

target = 0.001;
maxdays = 1000;

%% INITIALIZATION

clf
hold on
box on
colors = get(gca,'colororder');

%% COMPUTATION
```

```
[S,E,I,R,sinfty] = seir(R0,tL,tI,e0,i0,r0,target,maxdays);
days = length(I)-1;
new = S(1:days)-S(2:length(S));
new = [0;new];

%% OUTPUT

times = 0:days;

plot(times,S,'Color',colors(1,:),'LineWidth',1.4)
plot(times,E,'Color',colors(3,:),'LineWidth',1.4)
plot(times,I,'Color',colors(2,:),'LineWidth',1.4)
plot(times,R,'Color',colors(4,:),'LineWidth',1.4)

% use 'FontSize',18 in Octave
xlabel('days','FontSize',14)

% use 'FontSize',16 in Octave
ylabel('population fraction','FontSize',12)

legend('S','E','I','R','Location','West')

results = [times',S,E,I,R,new]
[M,J] = max(I);
maxI = M
maxday = J-1
finalS = sinfty
```

SEIR_comparison.m

```
%% SEIR\_comparison

% Plots a comparison of population classes for multiple scenarios

% Uses seir.m, 2020/11/27 version

% User specifies a list of values for one of the key parameters:
%   R0 is the basic reproductive number
%   tL is the mean incubation time
%   tI is the mean recovery time
%   i0 is the initial infectious fraction
%   r0 is the initial immune fraction
```

```
% e0 is calculated assuming exponential growth phase

% The program is designed so that only two lines need to be modified to
% make a new experiment (see '%%%' comments)
%     line 47 defines the independent variable values
%     line 79 links the independent variable name and values
% The legend is optional

% output figure:
%   left panel: S vs time
%   right panel: I vs time

% Change axis label font sizes for Octave

% by Glenn Ledder
% written 2020/11/27
% revised 2021/08/01

% direct comments to gledder@unl.edu

%% DEFAULT SCENARIO DATA

R0 = 2.5;
tL = 5;
tI = 10;
i0 = 0.001;
r0 = 0;

%% INDEPENDENT VARIABLE DATA

%%% This section needs to be modified for each experiment.

%%% xvals is the set of independent variable values
xvals = [5,4,3,2];

%% COMMON DATA

target = 0.001;
maxdays = 1000;

%% INITIALIZATION

clf
for k=1:2
```

```
    subplot(1,2,k)
    hold on
    box on
    % comment out next line if desired
    axis square

    % use 'FontSize',18 in Octave
    xlabel('days','FontSize',14)
end

N = length(xvals);
finalS = zeros(1,N);
maxI = zeros(1,N);
maxday = zeros(1,N);

%% COMPUTATION and PLOTS

for n=1:N

    %%% The left side of this statement needs to be the independent
    %%% variable for the experiment.
    R0 = xvals(n);

    eta = 1/tL;
    gamma = 1/tI;
    beta = gamma*R0;

    a = eta;
    b = eta-gamma+gamma*R0*i0;
    c = -gamma*R0*(1-i0-r0);
    b2 = b/2;
    rho = (sqrt(b2^2-a*c)-b2)/a;
    e0 = rho*i0;

    [S,~,I,~,sinfty] = seir(R0,tL,tI,e0,i0,r0,target,maxdays);
    days = length(I)-1;

    [M,J] = max(I);
    maxI(n) = M;
    maxday(n) = J-1;
    finalS(n) = sinfty;

    subplot(1,2,1)
    plot(0:days,S,'LineWidth',1.4)
    subplot(1,2,2)
```

```
    plot(0:days,I,'LineWidth',1.4)
end

subplot(1,2,1)
ylim([0,1])
% use 'FontSize',16 in Octave
ylabel('susceptible','FontSize',12)

subplot(1,2,2)
% use 'FontSize',16 in Octave
ylabel('infectious','FontSize',12)
legend('R0=5','R0=4','R0=3','R0=2','Location','Northeast')

%% OUTPUT

maxI = maxI
maxday = maxday
finalS = finalS
```

SEIR_paramstudy.m

```
%% SEIR parameter studies

% Runs an SEIR experiment to determine how maximum infectious fraction and
% final at-risk fraction depend on a parameter value

% Uses seir.m, 2020/11/27 version

% User specifies a list of values for one of the key parameters:
%   R0 is the basic reproductive number
%   tL is the mean incubation time
%   tI is the mean recovery time
%   i0 is the initial infectious fraction
%   r0 is the initial immune fraction

% e0 is calculated assuming exponential growth phase

% The program is designed so that only a few lines need to be modified to
% make a new experiment (see '%%%' comments)
%   lines 49-51 define the independent variable values
%   line 54 identifies the x axis label for the graph
%   line 86 links the independent variable name and values
```

```
% Output figure:
%   top left panel: max infectious fraction
%   top right panel: day of max new infections
%   bottom left panel: fraction of susceptibles who become infected
%   bottom right panel: final fraction susceptibles

% Change x axis label font size for Octave

% by Glenn Ledder
% written 2020/11/27
% revised 2021/08/01

% direct comments to gledder@unl.edu

%% DEFAULT SCENARIO DATA

R0 = 2.5;
tL = 5;
tI = 10;
i0 = 0.001;
r0 = 0;

%% INDEPENDENT VARIABLE DATA

%%% This section needs to be modified for each experiment.

%%% first and last are the min and max values for the independent variable
%%% N is the number of points (not subdivisions)
first = .1;
last = 6;
N = 60;

%%% xname is the name for the x axis label
xname = 'R0';

%% COMMON DATA

target = 0.001;
maxdays = 1000;

%% INITIALIZATION

clf
for k=1:4
    subplot(2,2,k)
```

```
    hold on
    box on

    % use 'FontSize',18 in Octave
    xlabel(xname,'FontSize',14)
end
colors = get(gca,'colororder');

% xvals holds whatever values are being used for the independent variable
xvals = linspace(first,last,N);

finalS = zeros(1,N);
fracI = zeros(1,N);
maxI = zeros(1,N);
maxday = zeros(1,N);

%% COMPUTATION

for n=1:N

    %%% The left side of this statement needs to be the independent
    %%% variable for the experiment.
    R0 = xvals(n);

    eta = 1/tL;
    gamma = 1/tI;
    beta = gamma*R0;

    a = eta;
    b = eta-gamma+gamma*R0*i0;
    c = -gamma*R0*(1-i0-r0);
    b2 = b/2;
    rho = (sqrt(b2^2-a*c)-b2)/a;
    e0 = rho*i0;
    s0 = 1-e0-i0-r0;

    [~,~,I,~,sinfty] = seir(R0,tL,tI,e0,i0,r0,target,maxdays);

    [M,J] = max(I);
    maxI(n) = M;
    maxday(n) = J-1;
    finalS(n) = sinfty;
    fracI(n) = (s0-sinfty)/s0;
end
```

```
%% OUTPUT

subplot(2,2,1)
plot(xvals,maxI,'Color',colors(3,:),'LineWidth',1.7)
ylabel('max fraction infectious')
subplot(2,2,2)
plot(xvals,maxday,'k','LineWidth',1.7)
ylabel('days for max infectious')
subplot(2,2,3)
plot(xvals,fracI,'Color',colors(2,:),'LineWidth',1.7)
ylabel('fraction infected')
subplot(2,2,4)
plot(xvals,finalS,'Color',colors(1,:),'LineWidth',1.7)
ylabel('final fraction susceptible')
```

References

1. Blackwood, J.C., Childs, L.M.: An introduction to compartmental modeling for the budding infectious disease modeler. *Letters in Biomathematics* (2018) doi: 10.1080/23737867.2018.1509026
2. Brauer, F.: Mathematical epidemiology is not an oxymoron. *BMC Public Health* (2009) doi: 10.1186/1471-2458-9-S1-S2
3. Davis, C.P.: *Swine Flu* https://www.medicinenet.com/swine_flu/article.htm. Cited 11 December 2020.
4. Diamond, J.: *Guns, Germs, and Steel*. W.W. Norton, New York (1997)
5. Gammack, D., Schaefer, E., Gaff, H.: Global dynamics emerging from local interactions: agent-based modeling for the life sciences, in *Mathematical Concepts and Methods in Modern Biology: Using Modern Discrete Models*, ed Robeva, R. & Hodge, T. Academic Press (2013)
6. Homp, M. and Ledder, G.: *The Mathematics of an Illness Outbreak: part 1, Disease Simulation Activity* (2019) https://drive.google.com/drive/folders/1k7ImUJEuO3hz5uSZ1hPVwv0KSm9aSEjs. Cited 29 November 2020.
7. Jungck, J.R., Gaff, H., Weisstein, A.E.: Mathematical manipulative models: In defense of "Beanbag Biology". *CBE-Life Sciences Education*, **9**, 201–211 (2010)
8. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. *Proc Royal Soc London*, **115**, 700–721 (1927) doi: 10.1098/rspa.1927.0118.
9. Khan, S.: *Derivative as a Concept* (2017) https://www.youtube.com/watch?v=N2PpRnFqnqY. Cited 29 November 2020.
10. Ledder, G.: *Mathematics for the Life Sciences: Calculus, Modeling, Probability, and Dynamical Systems*. Springer-Verlag, New York (2013)
11. McNeill, W.H.: *Plagues and Peoples*. Anchor Press, Garden City, NY (1976)
12. *Octave Online* https://octave-online.net/. Cited 11 December 2020.
13. Railsback, S.F., Grimm, V.: *Agent-based and Individual-based Modeling: A Practical Introduction*, 2nd edition. Princeton University Press, Princeton (2019)
14. Sanche, S., Lin, Y.T., Xu, C., Romero-Severson, E., Hengartner, N., Ke, R.: High contagiousness and rapid spread of severe respiratory syndrome coronavirus 2. *Emerging Infectious Diseases*, **26**: 1470–1477 (2020) doi: 10.3201/eid2607.200282.

15. Smith, G.: *Back of the Envelope Modelling of Infectious Disease Transmission Dynamics for Veterinary Students*. Cambridge Scholars Publishing, Newcastle-Upon-Tyne (2019).
16. Wikipedia: *Lotka-Volterra Equations*
https://en.wikipedia.org/wiki/Lotka-Volterra_equations. Cited 29 November 2020.
17. Wikipedia: *Moore's law*
https://en.wikipedia.org/wiki/Moore's_law. Cited 29 November 2020.

Index