

Mathematical Methods for Biology, Part 1

Dmitry Kondrashov

Table of contents

| | |
|---|-----------|
| Preface | 3 |
| modeling assumptions: theoretical and empirical | 3 |
| variables and parameters | 5 |
| units and dimensions | 6 |
| 1 One variable in discrete time | 8 |
| 1.1 Building dynamic models | 8 |
| 1.1.1 static population | 9 |
| 1.1.2 exponential population growth | 9 |
| 1.1.3 example with birth and death | 11 |
| 1.1.4 dimensions of birth and death rates | 11 |
| 1.1.5 general demographic model | 12 |
| 1.2 Solutions of linear difference models | 12 |
| 2 Plotting in Python | 17 |
| 2.0.1 arrays and basic plotting | 17 |
| 2.1 Numeric solutions of discrete models | 19 |
| 2.1.1 using for loops for iterative solutions of dynamic models | 19 |
| 2.1.2 plotting multiple curves with a legend | 21 |
| References | 24 |

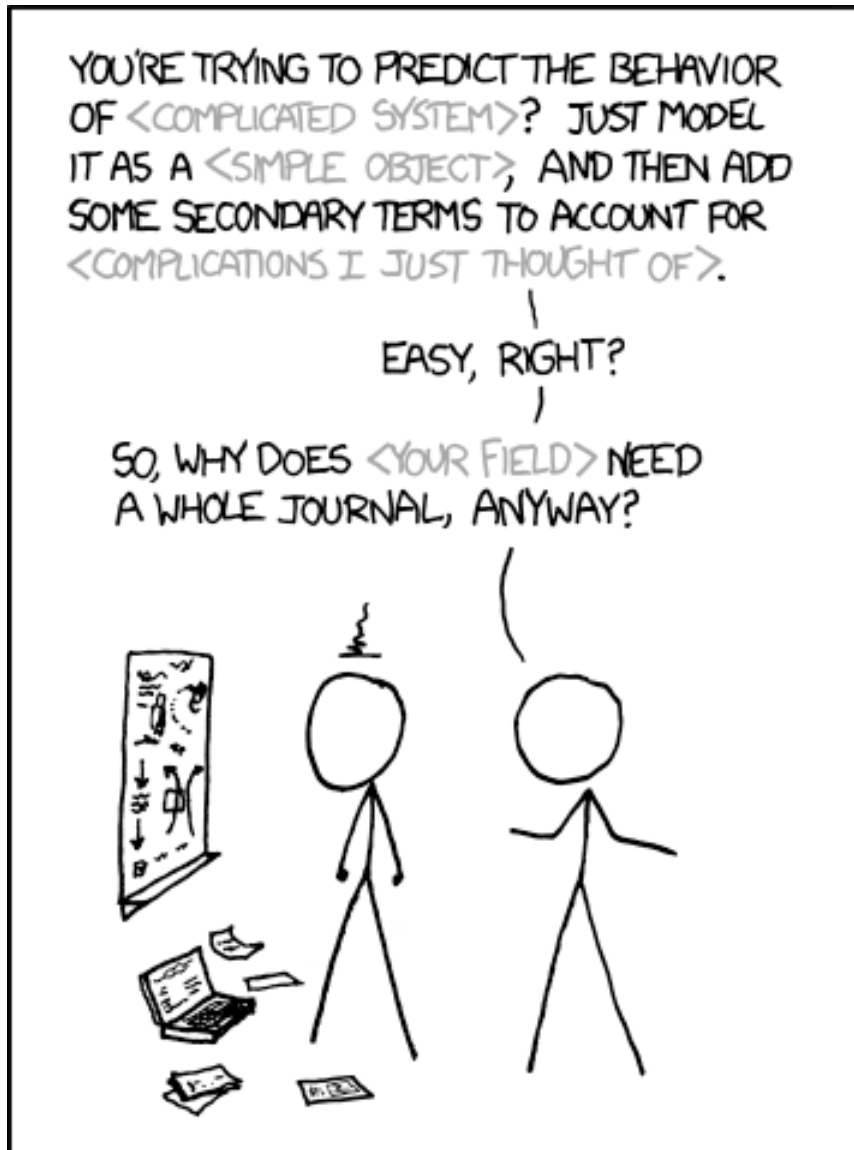
Preface

In this book you will find a collection of mathematical ideas, computational methods, and modeling tools for describing biological systems quantitatively. Biological science, like all natural sciences, is driven by experimental results. As with other sciences, there comes a point when accumulated data needs to be analyzed quantitatively, in order to formulate and test explanatory hypotheses. Biology has reached this stage, thanks to an explosion of data from molecular biology techniques, such as large-scale DNA sequencing, protein structure determination, data on gene regulatory networks, and signaling pathways. Quantitative skills have become necessary for anyone hoping to make sense of biological research.

Mathematical modeling necessarily involves making simplifying assumptions. Reality is generally too complex to be captured in a few equations, and this is especially true for living systems. Simplicity in modeling has at least two virtues: first, simple models can be grasped by our limited minds, and second, it allows for meaningful testing of the assumptions against the evidence. A complex model that fits the data may not provide any insights about how the system works, whereas a simple model which does not fit all the data can indicate where the assumptions break down. We will learn how to construct progressively more sophisticated models, beginning with the ridiculously simple.

modeling assumptions: theoretical and empirical

A mathematical model postulates a precise relationship between several quantities, attempting to mimic the behavior of a real system. All models rest on a set of assumptions, postulating how various quantities are interrelated. These assumptions generally come from two sources: a scientific theory, or experimental observations. For instance, a model of molecular motion may rest on the assumption that Newton's laws hold true. On the other hand, the observation that a drug injected into the bloodstream of a mammal is metabolized with an exponential time dependence is empirical. The benefit of models based on well-established theories, sometimes known as "first-principles models", is that they can be constructed without prior experimental knowledge of a particular system. Newton's laws apply to all sorts of classical mechanics objects, ranging in size from molecules to planets. Some prefer first-principles models, because they rely on well-established scientific principles, while others will argue that an empirical model more accurately reflects the behavior of the system at hand. From a mathematical standpoint, there is no difference between the two types of models. We will use the same tools to construct and analyze models, regardless of their origin.



LIBERAL-ARTS MAJORS MAY BE ANNOYING SOMETIMES, BUT THERE'S *NOTHING* MORE OBNOXIOUS THAN A PHYSICIST FIRST ENCOUNTERING A NEW SUBJECT.

Figure 0.1: Beware: a little knowledge of mathematical modeling can lead to arrogance.
<<http://xkcd.com/793/>>

A stated assumption can be written as a mathematical relationship, usually in the form of an equation relating quantities of interest. A postulated assumption may be expressed in words as “ X is proportional to Y ”, and can be written as the following equation: $X = aY$. Another model may postulate a relationship “ X is inversely proportional to the product of Y and Z ”, which is expressed as $X = a/YZ$.

Suppose we want to model the relationship between the height of individuals (H) and their weight (W). Measuring those quantities in some population results in the observation that the weight is proportional to the height, with an additive correction. Then we can write the following mathematical model, based on the empirical evidence: $W = aH + c$

In electricity, Ohm’s law governs the relationship between the flow of charged particles, called current (I), the electric potential (V) and the resistance of a conductor (R). This law states that the current through a conductor is proportional to the potential and inversely proportional to the resistance, and thus can be mathematically formulated:

$$I = \frac{V}{R}$$

variables and parameters

Mathematical models formulate relationships between different quantities that can be measured in real systems. There are two different types of quantities in models: *variables* and *parameters*. The same measurable quantity can be a variable or a parameter, depending on the role it plays in the model. A variable typically varies, either in time or in space, and the model tracks the changes in its value. On the other hand, a parameter typically usually stays the same for a particular manifestation of the model, e.g. an individual or a specific population. However, parameters can vary from individual to individual, or from population to population.

In the height and weight model above, the numbers H and W are the variables, which can change between different individuals. The parameters a and c can either be estimated from data for various subpopulations. Perhaps the values of the parameters are different for young people than for older people, or they are different for those who exercise regularly versus those who do not. Once the parameters have been set, one can predict W given H , or vice versa. Of course, since this is a model, it is only an approximation of reality. The deviations of predictions of the model from actual height or weight for an individual may tell us something interesting about the physiology of the individual.

There are three quantities in the equation for Ohm’s law, and the distinction between variables and parameters depends on the actual system that is being modeled. In order to distinguish between the two, consider which quantity is set prior to the experiment, and which one may vary over the course of the situation we are trying to model. For instance, if voltage is being applied to a material with constant resistance, and the potential may be varied, then V is the

independent variable, I is the dependent variable, and R is a parameter. On the other hand, if the setup uses a variable resistor (known as a potentiometer or pot), and the voltage remains constant, then V is a parameter, while I and R are variables. If both the voltage V and the resistance R can vary at the same time, then all three quantities are variables.

units and dimensions

Each variable and parameter has its own *dimension*, which describes the physical or biological meaning of the quantity. Examples are time, length, number of individuals, or concentration per time. It is important to distinguish the dimension of a quantity from the *units* of measurement. The same quantity can be measured in different units: length can be in meters or feet, population size can be expressed in individuals or millions of individuals. The value of a quantity depends on the units of measurement, but its essential dimensionality does not.

There is a fundamental requirement of mathematical modeling: all the terms in an equation must agree in dimensionality; e.g. time cannot be added to number of sheep, since this sum has no biological meaning. In order to express this rule, we will write the dimension of a quantity X as $[X]$. While X refers to a numerical value, $[X]$ describes its physical meaning. Then the above statement can be illustrated by the following example:

$$aX = bY^2 \Rightarrow [aX] = [bY^2]$$

In the equation $\$W = a H + c \$$ all the terms must have the dimension of weight, because that is the meaning of the left hand side of the equation. Therefore, c has the dimensions of weight as well. H of course has the dimension of length, so this implies that the parameter a has dimensions of weight divided by length. This can be summed up as follows:

$$[W] = [c] = \text{weight}; [H] = \text{length}; [a] = \frac{\text{weight}}{\text{length}}$$

While the dimensions are set by the equation, the units of these quantities can vary. Weight can be expressed in pounds, kilograms, or stones, and length can be represented in inches, meters, or light years.

The dimensions of current are defined to be the amount of charge moving per unit of time, and the dimensions of voltage are energy per unit of charge. This allows us to find the dimensions of resistance by the following basic algebra:

$$[V] = \frac{\text{energy}}{\text{charge}} = \frac{[I]}{[R]} = \frac{\text{charge/time}}{[R]} \Rightarrow [R] = \frac{\text{charge}^2}{\text{energy} * \text{time}}$$

Electric potential is measured in volts, and current in amperes. The standard unit of resistance is the Ohm, which is defined as one volt per ampere. But regardless of the choice of units, the dimensions of these quantities remains.

A quantity may be made *dimensionless* by expressing it in terms of particular *scale*. For instance, we can express the height of a person as a fraction of the mean height of the population. A tall person will have height expressed as a number greater than 1, and a short one will have height less than 1. Note that this dimensionless height has no units - they have been divided out by scaling the height by the mean height. In fact, the word dimensionless is somewhat misleading: while such quantities have no scale in the context of the algebraic relationship, a quantity retains its physical significance after rescaling: height expressed as a fraction of some chosen length still represents height. Nevertheless, the accepted term in dimensionless quantity, and we will stick with this convention. Later in the book we will learn how to use the technique of rescaling to simplify and analyze dynamic models.

1 One variable in discrete time

All living things change over time, and this evolution can be quantitatively measured and analyzed. Mathematics makes use of equations to define models that change with time, known as *dynamical systems*. In this unit we will learn how to construct models that describe the time-dependent behavior of some measurable quantity in life sciences. Numerous fields of biology use such models, and in particular we will consider changes in population size, the progress of biochemical reactions, the spread of infectious disease, and the spikes of membrane potentials in neurons, as some of the main examples of biological dynamical systems.

Many processes in living things happen regularly, repeating with a fairly constant time period. One common example is the reproductive cycle in species that reproduce periodically, whether once a year, or once an hour, like certain bacteria that divide at a relatively constant rate under favorable conditions. Other periodic phenomena include circadian (daily) cycles in physiology, contractions of the heart muscle, and waves of neural activity. For these processes, theoretical biologists use models with *discrete time*, in which the time variable is restricted to the integers. For instance, it is natural to count the generations in whole numbers when modeling population growth.

This chapter is devoted to analyzing dynamical systems in which time is measured in discrete steps. We will build dynamic models, find their mathematical solutions, and then use Python to compute the solutions and plot them. In this chapter you will learn to:

- build discrete-time models of populations using rate parameters
- define and verify mathematical solutions of these models
- use Python to compute and plot solutions

1.1 Building dynamic models

Let us construct our first models of biological systems! We will start by considering a population of some species, with the goal of tracking its growth or decay over time. The variable of interest is the number of individuals in the population, which we will call N . This is called the dependent variable, since its value changes depending on time; it would make no sense to say that time changes depending on the population size. Throughout the study of dynamical systems, we will denote the independent variable of time by t . To denote the population size at time t , we can write $N(t)$ but sometimes use N_t .

1.1.1 static population

In order to describe the dynamics, we need to write down a rule for how the population changes. Consider the simplest case, in which the population stays the same for all time. (Maybe it is a pile of rocks?) Then the following equation describes this situation:

$$N(t + 1) = N(t)$$

This equation mandates that the population at the next time step be the same as at the present time t . This type of equation is generally called a *difference equation*, because it can be written as a difference between the values at the two different times:

$$N(t + 1) - N(t) = 0$$

This version of the model illustrates that a difference equation at its core describes the *increments* of N from one time step to the next. In this case, the increments are always 0, which makes it plain that the population does not change from one time step to the next.

1.1.2 exponential population growth

Let us consider a more interesting situation: as a colony of dividing bacteria, such as *E. coli*, shown in {numref}fig-cell-div. We assume that each bacterial cell divides and produces two daughter cells at fixed intervals of time, and let us further suppose that bacteria never die. Essentially, we are assuming a population of immortal bacteria with clocks. This means that after each cell division the population size doubles. As before, we denote the number of cells in each generation by $N(t)$, and obtain the equation describing each successive generation:

$$N(t + 1) = 2N(t)$$

It can also be written in the difference form, as above:

$$N(t + 1) - N(t) = N(t)$$

The increment in population size is determined by the current population size, so the population in this model is forever growing. This type of behavior is termed *exponential growth* and we will see how to express the solution algebraically in the next section.

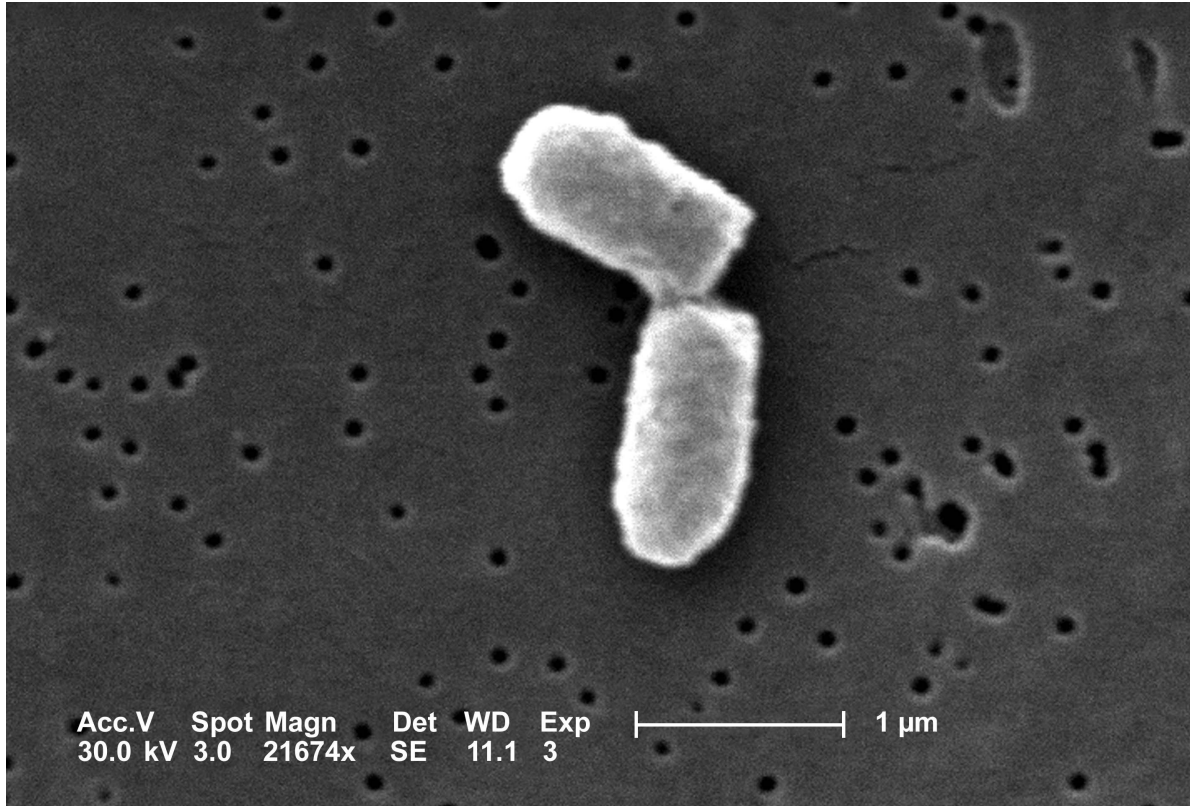


Figure 1.1: Scanning electron micrograph of a dividing *Escherichia coli* bacteria (image by Evangeline Sowers, Janice Haney Carr (CDC) in public domain via Wikimedia Commons)

1.1.3 example with birth and death

Suppose that a type of fish lives to reproduce only once after a period of maturation, after which the adults die. In this simple scenario, half of the population is female, a female always lays 1000 eggs, and of those, 1% survive to maturity and reproduce. Let us set up the model for the population growth of this idealized fish population. The general idea, as before, is to relate the population size at the next time step $N(t+1)$ to the population at the present time $N(t)$.

Let us tabulate both the increases and the decreases in the population size. We have $N(t)$ fish at the present time, but we know they all die after reproducing, so there is a decrease of $N(t)$ in the population. Since half of the population is female, the number of new offspring produced by $N(t)$ fish is $500N(t)$. Of those, only 1% survive to maturity (the next time step), and the other 99% ($495N(t)$) die. We can add all the terms together to obtain the following difference equation:

$$N(t+1) = N(t) - N(t) + 500N(t) - 495N(t) = 5N(t)$$

The number 500 in the expression is the *birth rate* of the population per individual, and the negative terms add up to the *death rate* of 496 per individual. We can re-write the equation in difference form:

$$N(t+1) - N(t) = 4N(t)$$

This expression again generates growth in the population, because the birth rate outweighs the death rate. (allman_mathematical_2003?)

1.1.4 dimensions of birth and death rates

What distinguishes a mathematical model from a mathematical equation is that the quantities involved have a real-world meaning. Each quantity represents a measurement, and associated with each one are the *units* of measurement, which are familiar from science courses. In addition to units, each variable and parameter has a meaning, which is called the *dimension* of the quantity. For example, any measurement of length or distance has the same dimension, although the units may vary. The value of a quantity depends on the units of measurement, but its essential dimensionality does not. One can convert a measurement in meters to that in light-years or cubits, but one cannot convert a measurement in number of sheep to seconds - that conversion has no meaning.

Thus leads us to the fundamental rule of mathematical modeling: **terms that are added or subtracted must have the same dimension**. This gives mathematical modelers a useful tool called *dimensional analysis*, which involves replacing the quantities in an equation with

their dimensions. This serves as a check that all dimensions match, as well as allowing to deduce the dimensions of any parameters for which the dimension was not specified.

In the case of population models, the birth and death rates measure the number of individuals that are born (or die) within a reproductive cycle for every individual at the present time. Their dimensions must be such that the terms in the equation all match:

$$[N(t+1) - N(t)] = [\textit{population}] = [r][N(t)] = [r] * [\textit{population}]$$

This implies that $[r]$ is algebraically dimensionless. However, the meaning of r is the rate of change of population over one (generation) time step. r is the birth or death rate of the population *per generation*, which is what makes it dimensionless. If the length of the generation were to change, but the reproduction and death per generation remain the same, then the parameter r would be the same, because it had been *rescaled* by the length of the generation. If they were to be reported in *absolute* units (e.g. individuals per year) then the rate would be different.

1.1.5 general demographic model

We will now write a general difference equation for any population with constant birth and death rates. This will allow us to substitute arbitrary values of the birth and death rates to model different biological situations. Suppose that a population has the birth rate of b per individual, and the death rate d per individual. Then the general model of the population size is:

$$N(t+1) = (1 + b - d)N(t)$$

(lin-pop)

The general equation also allows us to check the dimensions of birth and death rates, especially as written in the incremental form: $N(t+1) - N(t) = (b - d)N(t)$. The change in population rate over one reproductive cycle is given by the current population size multiplied by the difference of birth and death rates, which as we saw are algebraically dimensionless. The right hand side of the equation has the dimensions of population size, matching the difference on the left hand side. (edelstein-keshet_mathematical_2005?)

1.2 Solutions of linear difference models

We saw in the last section that we can write down equations to describe, step by step, how a variable changes over time. Let us define what the terminology of these equations:

“{admonition} Definition An equation to describe a variable (e.g. N) that changes over discrete time steps described by the integer variable t is called a *difference equation* or a *discrete-time dynamic model*. These equations can be written in two ways, either in *recurrent form*:

$$N(t+1) = f(N(t))$$

(recur-eq)

or in *increment form*:

$$N(t+1) - N(t) = g(N(t))$$

(recur-eq)

simple linear difference models

Having set up the difference equation models, we would naturally like to solve them to find

{admonition} Definition

A function $N(t)$ is a *solution* of a difference equation $N(t+1) = f(N(t))$ if it satisfies

For instance, let us take our first model of the static population, $N(t+1) = N(t)$. Any constant function is a solution, for example, $N(t) = 0$, or $N(t) = 10$. There are actually as many solutions as there are numbers, that is, infinitely many! In order to specify exactly what happens in the model, we need to specify the size of the population at some point, usually, at the “beginning of time”, $t = 0$. This is called the *initial condition* for the model, and for a well-behaved difference equation it is enough to determine a unique solution. For the static model, specifying the initial condition is the same as specifying the population size for all time.

Now let us look at the general model of population growth with constant birth and death rates. We saw in equation {eq}lin-pop above that these can be written in the form $N(t+1) = (1+b-d)N(t)$. To simplify, let us combine the numbers into one growth parameter $r = 1+b-d$, and write down the general equation for population growth with constant growth rate:

$$N(t+1) = rN(t)$$

(lin-pop-r)

To find the solution, consider a specific example, where we start with the initial population size $N_0 = 1$, and the growth rate $r = 2$. The sequence of population sizes is: 1, 2, 4, 8, 16, etc. This is described by the formula $N(t) = 2^t$.

In the general case, each time step the solution is multiplied by r , so the solution has the same exponential form. The initial condition N_0 is a multiplicative constant in the solution, and one can verify that when $t = 0$, the solution matches the initial value:

$$N(t) = r^t N_0$$

(lin-pop-sol)

I would like the reader to pause and consider this remarkable formula. No matter what the birth and death parameters are selected, this solution predicts the population size at any point in time t .

In order to verify that the formula for $N(t)$ is actually a solution in the meaning of definition, we need to check that it actually satisfies the difference equation for all t , not just a few time steps. This can be done algebraically by plugging in $N(t+1)$ into the left side of the dynamic model and $N(t)$ into the right side and checking whether they match. For $N(t)$ given by equation {eq}lin-pop-sol, $N(t+1) = r^{t+1}N_0$, and thus the dynamic model becomes:

$$r^{t+1}N_0 = r \times r^t N_0$$

Since the two sides match, this means the solution is correct.

The solutions in equation {eq}lin-pop-sol are exponential functions, which have a limited menu of behaviors, depending on the value of r . If $r > 1$, multiplication by r increases the size of the population, so the solution $N(t)$ will grow (see {numref}fig-exp-growth. If $r < 1$, multiplication by r decreases the size of the population, so the solution $N(t)$ will decay (see {numref}fig-exp-decay). Finally, if $r = 1$, multiplication by r leaves the population size unchanged, like in the pile of rocks model. Here is the complete classification of the behavior of population models with constant birth and death rates (assuming $r > 0$):

“{admonition} Classification of solutions of linear dynamic models :class: tip

For a difference equation $N(t+1) = rN(t)$, solutions can behave in one of three ways:

- $|r| > 1$: $N(t)$ grows without bound
- $|r| < 1$: $N(t)$ decays to 0
- $|r| = 1$: the absolute value of $N(t)$ remains constant

See examples of graphs of solutions of such equations with r greater than 1 in {numref}fig-

{figure} images/ch1_exp_growth.png

name: fig-exp-growth

Growth of a population that doubles every generation over 6 generations.

“{figure} images/ch1_exp_decay.png
name: fig-exp-decay

Decay of a population in which half the individuals die every time step over 6 generations

linear difference models with a constant term

Now let us consider a dynamic model that combines two different rates: a proportional rate (r) and a constant rate (a).

$$N(t+1) = rN(t) + a$$

The right-hand-side of this equation is a linear function of N , so this is a linear difference equation. The pesky constant term a is the problem.

$$N(t+1) \neq r^t N_0 + a$$

To solve it, we need to try a different form: specifically, an exponential with an added constant.

$$N(t+1) = c_1 r^{t+1} + c_2 = rN(t) + a = rc_1 r^t + rc_2 + a$$

This equation has the same term $c_1 r^{t+1}$ on both sides, so they can be subtracted out.

$$N(t) = c r^t + \frac{a}{1-r}$$

`\label{eq:ch14_sol_wconst}`

Example. Take the difference equation $N(t+1) = 0.5 N(t) + 40$ with initial value $N(0) = 20$.

$$N(t+1) = 20 \times 0.5^{t+1} + 80 = 0.5 \times (20 \times 0.5^t + 80) + 40 = 20 \times 0.5^t + 80$$

The equation is satisfied and therefore the solution is correct.

`<!-- quarto-file-metadata: eyJyZXNvdXJjZURpciI6Ii4ifQ== -->`{=html}`

``{=html}`

`<!-- quarto-file-metadata: eyJyZXNvdXJjZURpciI6Ii4iLCJib29rSXRlbVR5cGUiOiJjaGFwdGVyIiwiaYm9va`

2 Plotting in Python

You can find an introduction to the plotting library [matplotlib here](#).

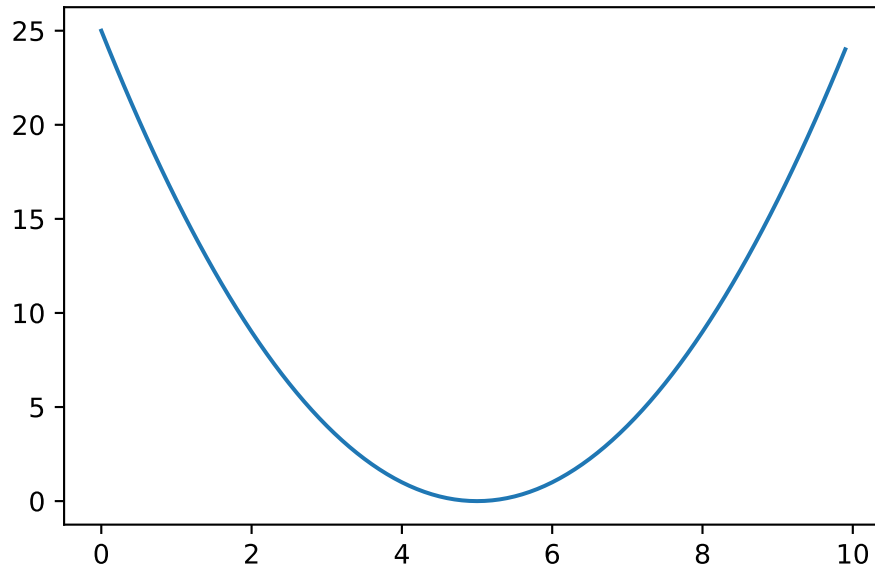
```
# Import packages
import numpy as np # package for work with arrays and matrices
import matplotlib.pyplot as plt # package with plotting capabilities
```

2.0.1 arrays and basic plotting

Here is an example of performing calculations with arrays (vectors) of values and plotting the results:

```
x = np.arange(0,10,0.1) # create an array of numbers between 0 and 10 with step 0.1
print(np.shape(x))
y = (x-5)**2 # do calculations on all the array values, call it y
plt.plot(x,y) # plot x vs y
plt.show()
```

(100,)



A two dimensional array (matrix) can be defined as follows, and the function `np.shape` prints out the number of rows and columns in the matrix:

```
x = np.array([[1,2,3],[4,5,6]])  
print(np.shape(x))  
print(x)
```

```
(2, 3)  
[[1 2 3]  
 [4 5 6]]
```

An example of concatenating a text string together with a numeric variable, which can then be used for labels or legends in plots:

```
prob = 0.5  
string1 = 'The value of prob is ' + str(prob)  
print(string1)
```

```
The value of prob is 0.5
```

2.1 Numeric solutions of discrete models

Difference equations, as we saw above, can be written in the form of $x_{t+1} = f(x_t)$. At every step, the model takes the current value of the dependent variable x_t , feeds it into the function $f(x)$, and takes the output as the next value x_{t+1} . The same process repeats every iteration, which is why difference equations written in this form are called *iterated maps*.

Computers are naturally suited for precise, repetitive operations. In our first example of a computational algorithm, we will iterate a given function to produce a sequence of values of the dependent variable x . We only need two things: to specify a computer function $f(x)$, which returns the value of the iterated map for any input value x , and the initial value x_0 . Then it is a matter of repeating the operation of evaluating $f(x_t)$ and storing it as the next value x_{t+1} . Below is the pseudocode for the algorithm. Note that I will use arrows to indicate variable assignment, square brackets `[]` for indexing of vector, and start indexing at 0, consistent with python convention.

Iterative solution of difference equations:

- define the iterated map function $F(x)$
- set N to be the number of iterations (time steps)
- set the initial condition x_0
- initialize array x with initial value x_0
- for i from 0 to $N - 1$

$$- x[i + 1] \leftarrow F(x[i])$$

The resulting sequence of values $x_0, x_1, x_2, \dots, x_N$ is called a *numeric solution* of the given difference equation. It has two disadvantages compared to an analytic solution: first, the solution can only be obtained for a specific initial value and number of iterations, and second, any computer simulation inevitably introduces some errors, for instance from round-off. In practice, however, most complex dynamical systems have to be solved numerically, as analytical solutions are difficult or impossible to find.

2.1.1 using for loops for iterative solutions of dynamic models

Here is a generic linear demographic model

$$x(t + 1) = x(t) + bx(t) - dx(t) = rx(t)$$

Example of a script for producing a numeric solution of a discrete time dynamic model:

```
numsteps = 20 # number of iterations
birth = 0.8 # birth rate
```

```

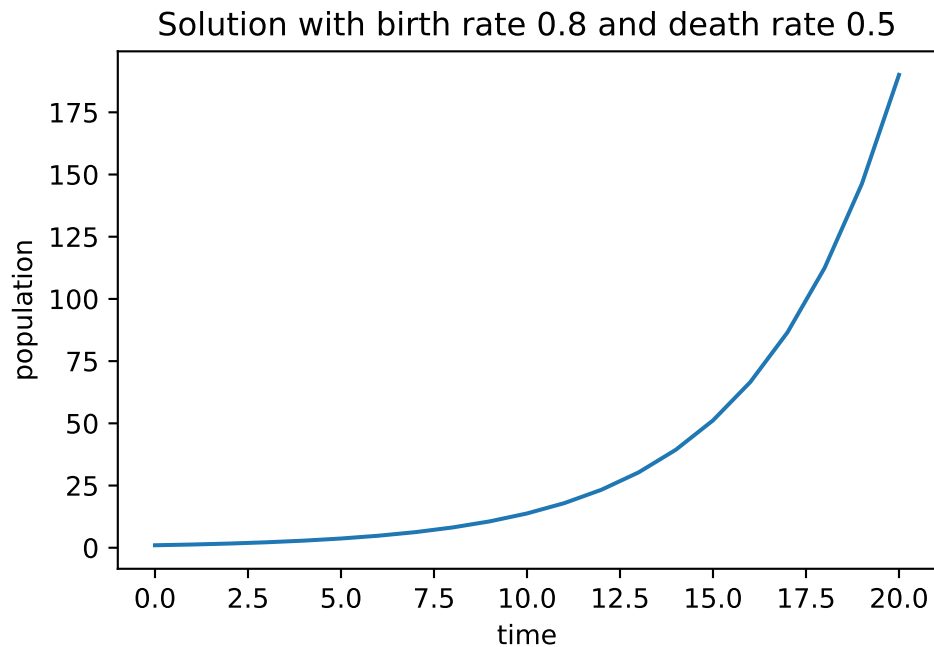
death = 0.5 # death rate
pop = np.zeros(numsteps+1) # initialize solution array
pop[0] = 1 # initial value
t = np.arange(numsteps+1) # initialize time vector
print(t)

for i in range(numsteps):
    pop[i+1] = pop[i] + birth*pop[i] - death*pop[i]# linear demographic model

plt.plot(t, pop) # plot solution
plt.xlabel('time')
plt.ylabel('population')
title = 'Solution with birth rate ' + str(birth) + ' and death rate ' + str(death)
plt.title(title)
plt.show()

```

[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]



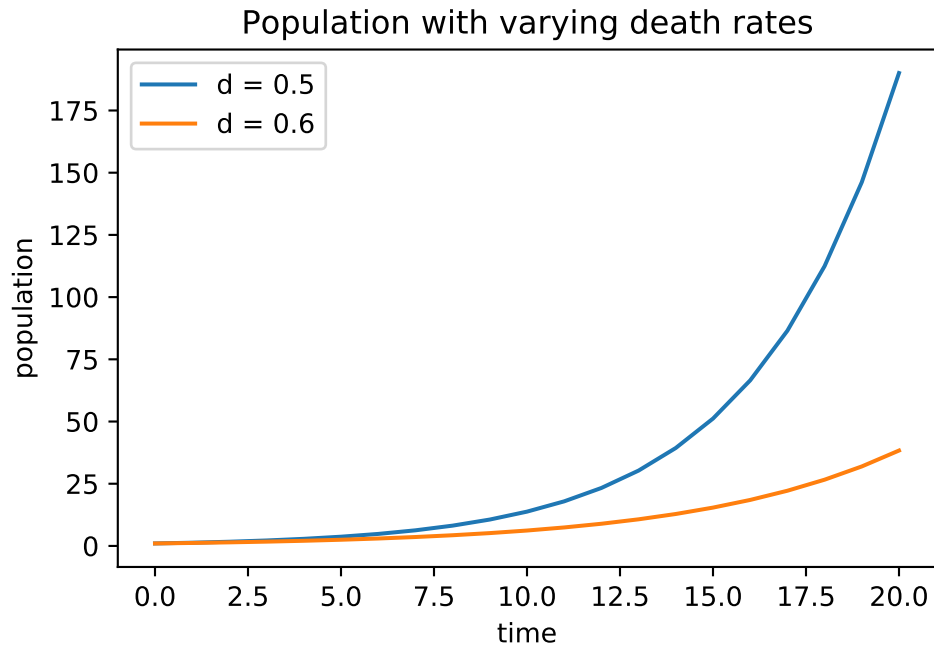
2.1.2 plotting multiple curves with a legend

Multiple solution plots can be overlaid on the same figure, as long as the `plt.show()` is only used once in the end. For multiple graphs it's best to use multiple colors and a legend to label different curves, using the option `label` in the `plt.plot` function and adding the function `plt.legend()` before producing the figure. Here's an example with solutions of the demographic model with different death rates:

```
numsteps = 20 # number of iterations
birth = 0.8 # birth rate
death = 0.5 # death rate
pop = np.zeros(numsteps+1) # initialize solution array
pop[0] = 1 # initial value
t = np.arange(numsteps+1) # initialize time vector
for i in range(numsteps):
    pop[i+1] = pop[i] + birth*pop[i] - death*pop[i] # linear demographic model

plt.plot(t, pop, label = 'd = '+str(death)) # plot solution
plt.xlabel('time')
plt.ylabel('population')
death = 0.6 # death rate
pop = np.zeros(numsteps+1) # initialize solution array
pop[0] = 1 # initial value
t = np.arange(numsteps+1) # initialize time vector
for i in range(numsteps):
    pop[i+1] = pop[i] + birth*pop[i] - death*pop[i] # linear demographic model
plt.plot(t, pop, label = 'd = '+str(death)) # plot solution

title = 'Population with varying death rates'
plt.title(title)
plt.legend()
plt.show()
```



n ### random number generators

Numpy provides a variety of random number generators, and we'll use these functions in the course for many purposes. Here is an example of producing arrays of random normally distributed numbers. The function requires inputs of the mean, the standard deviation, and the number of random values (or size of the array):

```
mu = 5

sigma = 0.5

num = 30

norm_sample = np.random.normal(mu, sigma, num)

print(norm_sample)

print("The mean of the sample is " + str(np.mean(norm_sample)))

print("The standard deviation of the sample is " + str(np.std(norm_sample)))
```

```
[4.40396996 6.40556978 4.85521397 4.86106979 6.1586708 5.11719756
```

```
4.53072189 5.81609864 4.74994362 4.717882    5.86602804 5.6235151
4.96510026 5.99310193 4.78383587 4.64535289 5.19321571 5.75845096
5.30639614 4.4573524  5.1313234  5.1774245  4.85275266 4.52176585
4.98496096 4.74024936 6.15001186 5.47509588 4.73988724 4.69721766]
The mean of the sample is 5.155979222673656
The standard deviation of the sample is 0.559759791448936
```

References