

4.8 PROJECTS

Project 4.1 Parasitic relationships

For this project, we assume that you have read the material on discrete difference equations in Section 4.4.

A parasite is an organism that lives either on or inside another organism for part of its life. A parasitoid is a parasitic organism that eventually kills its host. A parasitoid insect infects a host insect by laying eggs inside it then, when these eggs later hatch into larvae, they feed on the live host. (Cool, huh?) When the host eventually dies, the parasitoid adults emerge from the host body.

The *Nicholson-Bailey model*, first proposed by Alexander Nicholson and Victor Bailey in 1935 [43], is a pair of difference equations that attempt to simulate the relative population sizes of parasitoids and their hosts. We represent the size of the host population in year t with $H(t)$ and the size of the parasitoid population in year t with $P(t)$. Then the difference equations describing this model are

$$\begin{aligned} H(t) &= r \cdot H(t-1) \cdot e^{-aP(t-1)} \\ P(t) &= c \cdot H(t-1) \cdot (1 - e^{-aP(t-1)}) \end{aligned}$$

where

- r is the average number of surviving offspring from an uninfected host,
- c is the average number of eggs that hatch inside a single host, and
- a is a scaling factor describing the searching efficiency or search area of the parasitoids (higher is more efficient).

The value $(1 - e^{-aP(t-1)})$ is the probability that a host is infected when there are $P(t-1)$ parasitoids, where e is Euler's number (the base of the natural logarithm). Therefore,

$$H(t-1) \cdot (1 - e^{-aP(t-1)})$$

is the number of hosts that are infected during year $t-1$. Multiplying this by c gives us $P(t)$, the number of new parasitoids hatching in year t . Notice that the probability of infection grows exponentially as the size of the parasitoid population grows. A higher value of a also increases the probability of infection.

Question 4.1.1 Similarly explain the meaning of the difference equation for $H(t)$. ($e^{-aP(t-1)}$ is the probability that a host is not infected.)

Part 1: Implement the model

To implement this model, write a function

```
NB(hostPop, paraPop, r, c, a, years)
```

that uses these difference equations to plot both population sizes over time. Your function should plot these values in two different ways (resulting in two different plots). First, plot the host population size on the x -axis and the parasitoid population size on the y -axis. So each point represents the two population sizes in a particular year. Second, plot both population sizes on the y -axis, with time on the x -axis. To show both population sizes on the same plot, call the `pyplot.plot` function for each population list before calling `pyplot.show`. To label each line and include a legend, see the end of Section 4.2.

Question 4.1.2 Write a `main` function that calls your `NB` function to simulate initial populations of 24 hosts and 12 parasitoids for 35 years. Use values of $r = 2$, $c = 1$, and $a = 0.056$. Describe and interpret the results.

Question 4.1.3 Run the simulation again with the same parameters, but this time assign a to be `-math.log(0.5) / paraPop`. (This is $a = -\ln 0.5/12 \approx 0.058$, just slightly above the original value of a .) What do you observe?

Question 4.1.4 Run the simulation again with the same parameters, but this time assign $a = 0.06$. What do you observe?

Question 4.1.5 Based on these three simulations, what can you say about this model and its sensitivity to the value of a ?

Part 2: Constrained growth

An updated Nicholson-Bailey model incorporates a *carrying capacity* that keeps the host population under control. The carrying capacity of an ecosystem is the maximum number of organisms that the ecosystem can support at any particular time. If the population size exceeds the carrying capacity, there are not enough resources to support the entire population, so some individuals do not survive.

In this revised model, $P(t)$ is the same, but $H(t)$ is modified to be

$$H(t) = H(t-1) \cdot e^{-aP(t-1)} \cdot e^{r(1-H(t-1)/K)}$$

where K is the carrying capacity. In this new difference equation, the average number of surviving host offspring, formerly r , is now represented by

$$e^{r(1-H(t-1)/K)}.$$

Notice that, when the number of hosts $H(t-1)$ equals the carrying capacity K , the exponent equals zero. So the number of surviving host offspring is $e^0 = 1$. In general, as the number of hosts $H(t-1)$ gets closer to the carrying capacity K , the exponent gets smaller and the value of the expression above gets closer to 1. At the other extreme, when $H(t-1)$ is close to 0, the expression is close to e^r . So, overall, the number of surviving offspring varies between 1 and e^r , depending on how close $H(t-1)$ comes to the carrying capacity.

Write a function

```
NB_CC(hostPop, paraPop, r, c, a, K, years)
```

that implements this modified model and generates the same plots as the previous function.

Question 4.1.6 *Call your NB_CC function to simulate initial populations of 24 hosts and 12 parasitoids for 35 years. Use values of $r = 1.5$, $c = 1$, $a = 0.056$, and $K = 40$. Describe and interpret the results.*

Question 4.1.7 *Run your simulation with all three values of a that we used in Part 1. How do these results differ from the prior simulation?*

Project 4.2 Financial calculators

In this project, you will write three calculators to help someone (maybe you) plan for your financial future.

Part 1: How long will it take to repay my college loans?

Your first calculator will compare the amount of time it takes to repay student loans with two different monthly payment amounts. Your main program will need to prompt for

- the student loan balance at the time repayment begins
- the nominal annual interest rate (the monthly rate times twelve)
- two monthly payment amounts

Then write a function

```
comparePayoffs(amount, rate, monthly1, monthly2)
```

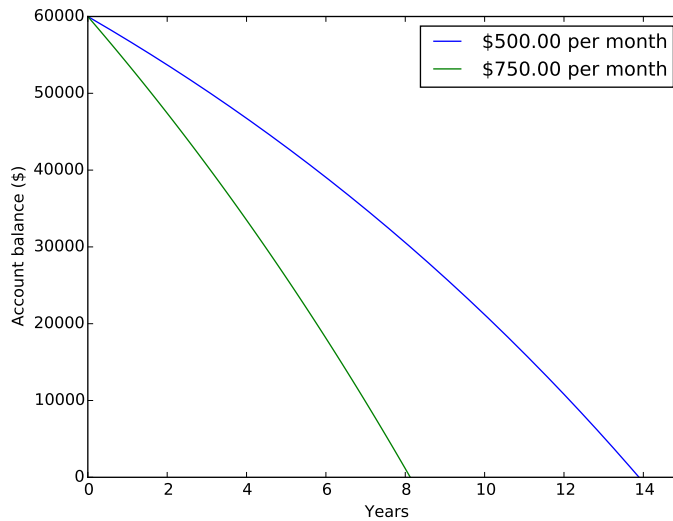
that computes the number of months required to pay off the loan with each monthly payment amount. The interest on the loan balance should compound monthly at a rate of `rate / 100 / 12`. Your function should also plot the loan balances, with each payment amount, over time until both balances reach zero. Then it should print the length of both repayment periods and how much sooner the loan will be paid off if the higher monthly payment is chosen. For example, your output might look like this:

```
Initial balance: 60000
Nominal annual percentage rate: 5
Monthly payment 1: 500
Monthly payment 2: 750
```

```
If you pay $500.00 per month, the repayment period will be 13 years
and 11 months.
```

```
If you pay $750.00 per month, the repayment period will be 8 years
and 2 months.
```

```
If you pay $250.00 more per month, you will repay the loan 5 years
and 9 months earlier.
```



Question 4.2.1 How long would it take to pay off \$20,000 in student loans with a 4% interest rate if you paid \$100 per month? Approximately how much would you have to pay per month to pay off the loan in ten years?

Question 4.2.2 If you run your program to determine how long it would take to pay off the same loan if you paid only \$50 per month, you should encounter a problem. What is it?

Part 2: How much will I have for retirement?

Your second calculator will compare the retirement nest eggs that result from making two different monthly investments in a retirement fund. Your program should prompt for the following inputs:

- the initial balance in the retirement account
- the current age of the investor
- the desired retirement age of the investor
- the expected nominal annual rate of return on the investment
- two monthly investment amounts

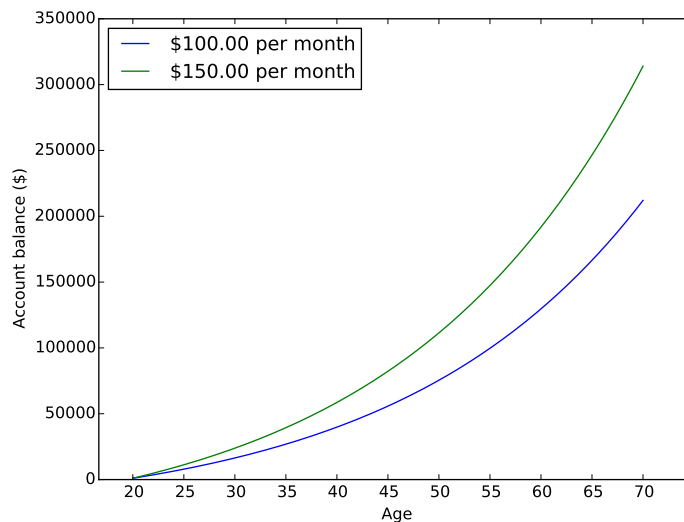
Then write a function

```
compareInvestments(balance, age, retireAge, rate, monthly1, monthly2)
```

that computes the final balance in the retirement account, for each monthly investment, when the investor reaches his or her retirement age. The interest on the current balance should compound monthly at a rate of $\text{rate} / 100 / 12$. The function should plot the growth of the retirement account balance for both monthly investment amounts, and then print the two final balances along with the additional amount that results from the higher monthly investment. For example, your output might look like this:

Initial balance: 1000
 Current age: 20
 Retirement age: 70
 Nominal annual percentage rate of return: 4.2
 Monthly investment 1: 100
 Monthly investment 2: 150

The final balance from investing \$100.00 per month: \$212030.11.
 The final balance from investing \$150.00 per month: \$313977.02.
 If you invest \$50.00 more per month, you will have \$101946.91 more at retirement.



Question 4.2.3 Suppose you are 30 and, after working for a few years, have managed to save \$6,000 for retirement. If you continue to invest \$200 per month, how much will you have when you retire at age 72 if your investment grows 3% per year? How much more will you have if you invest \$50 more per month?

Part 3: How long will my retirement savings last?

Your third calculator will initially perform the same computation as your previous calculator, but with only one monthly investment amount. After it computes the final balance in the account, it will estimate how long that nest egg will last into retirement. Your program will need to prompt for the same values as above (but only one monthly investment amount), plus the percentage of the final balance the investor plans to withdraw in the first year after retirement. Then write a function

`retirement(amount, age, retireAge, rate, monthly, percentWithdraw)`

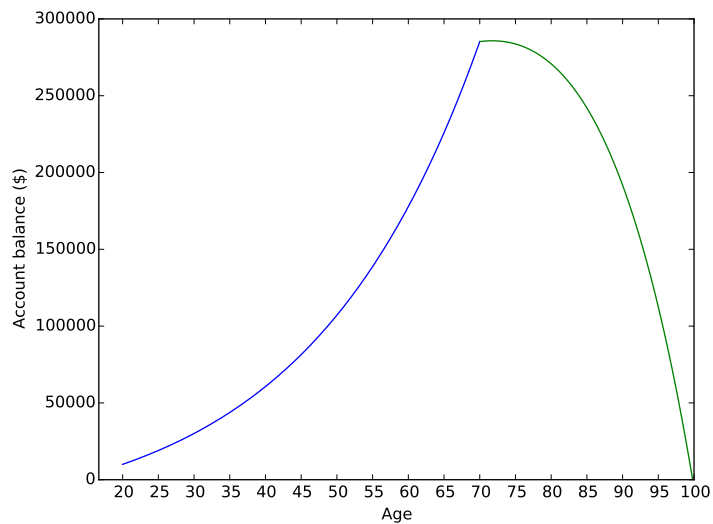
that adds the monthly investment amount to the balance in the retirement account until the investor reaches his or her retirement age, and then, after retirement age, withdraws a monthly amount. In the first month after retirement, this amount should be one-twelfth of `percentWithdraw` of the current balance. For every month thereafter, the withdrawal amount should increase according to the rate of inflation,

P4.2-4 ■ Discovering Computer Science, Second Edition

assumed to be 3% annually. Every month, the interest on the current balance should compound at a rate of $\text{rate} / 100 / 12$. The function should plot the retirement account balance over time, and then print the age at which the retirement funds run out. For example, your program output might look like this:

```
Initial balance: 10000
Current age: 20
Retirement age: 70
Annual percentage rate of return: 4.2
Monthly investment: 100
Annual withdrawal % at retirement: 4
```

Your savings will last until you are 99 years and 10 months old.



Question 4.2.4 How long will your retirement savings last if you follow the plan outlined in Question 4.2 (investing \$200 per month) and withdraw 4% at retirement?

***Project 4.3 Market penetration**

For this project, we assume that you have read Section 4.4.

In this project, you will write a program that simulates the adoption of a new product in a market over time (i.e., the product's "market penetration") using a difference equation developed by marketing researcher Frank Bass in 1969 [5]. The *Bass diffusion model* assumes that there is one product and a fixed population of N eventual adopters of that product. In the difference equation, $A(t)$ is the fraction (between 0 and 1) of the population that has adopted the new product t weeks after the product launch. Like the difference equations in Section 4.4, the value of $A(t)$ will depend on the value of $A(t - \Delta t)$, the fraction of the population that had adopted the product at the previous time step, $t - \Delta t$. Since t is measured in weeks, Δt is some fraction of a week. The rate of product adoption depends on two factors.

1. The segment of the population that has not yet adopted the product adopts it at a constant adoption rate of r due to chance alone. By the definition of $A(t)$, the fraction of the population that *had* adopted the product at the previous time step is $A(t - \Delta t)$. Therefore, the fraction of the population that *had not yet* adopted the product at the previous time step is $1 - A(t - \Delta t)$. So the fraction of new adopters during week t from this constant adoption rate is

$$r \cdot (1 - A(t - \Delta t)) \cdot \Delta t.$$

2. Second, the rate of adoption is affected by word of mouth within the population. Members of the population who have already adopted the product can influence those who have not. The more adopters there are, the more potential interactions exist between adopters and non-adopters, which boosts the adoption rate. The fraction of all potential interactions that are between adopters and non-adopters in the previous time step is

$$\underbrace{A(t - \Delta t)}_{\text{fraction that are adopters}} \cdot \underbrace{(1 - A(t - \Delta t))}_{\text{fraction that are not adopters}}.$$

The fraction of these interactions that result in a new adoption during one week is called the *social contagion*. We will denote the social contagion by s . So the fraction of new adopters due to social contagion during the time step ending at week t is

$$s \cdot A(t - \Delta t) \cdot (1 - A(t - \Delta t)) \cdot \Delta t.$$

The social contagion measures how successfully adopters are able to convince non-adopters that they should adopt the product. At the extremes, if $s = 1$, then every interaction between a non-adopter and an adopter results in the non-adopter adopting the product. On the other hand, if $s = 0$, then the current adopters cannot convince any non-adopters to adopt the product.

Putting these two parts together, the difference equation for the Bass diffusion model is

$$A(t) = A(t - \Delta t) + \underbrace{r \cdot (1 - A(t - \Delta t)) \cdot \Delta t}_{\text{fraction of new adopters from constant rate}} + \underbrace{s \cdot A(t - \Delta t) \cdot (1 - A(t - \Delta t)) \cdot \Delta t}_{\text{fraction of new adopters from social contagion}}$$

Part 1: Implement the Bass diffusion model

To implement the Bass diffusion model, write a function

```
productDiffusion(chanceAdoption, socialContagion, weeks, dt)
```

The parameters `chanceAdoption` and `socialContagion` are the values of r and s , respectively. The last two parameters give the number of `weeks` to simulate and the value of Δt to use. Your function should plot two curves, both with time on the x -axis and the proportion of adopters on the y -axis. The first curve is the total fraction of the population that has adopted the product by time t . This is $A(t)$ in the difference equation above. The second curve will be the rate of change of $A(t)$. You can calculate this by the formula

$$\frac{A(t) - A(t - \Delta t)}{\Delta t}.$$

Equivalently, this rate of change can be thought of as the fraction of new adopters at any time step, normalized to a weekly rate, i.e., the fraction of the population added in that time step divided by `dt`.

Write a program that uses your function to simulate a product launch over 15 weeks, using $\Delta t = 0.01$. For this product launch, we will expect that adoption of the product will move slowly without a social effect, but that social contagion will have a significant impact. To model these assumptions, use $r = 0.002$ and $s = 1.03$.

Question 4.3.1 Describe the picture and explain the pattern of new adoptions and the resulting pattern of total adoptions over the 15-week launch.

Question 4.3.2 Now make r very small but leave s the same ($r = 0.00001$, $s = 1.03$), and answer the same question. What kind of market does this represent?

Question 4.3.3 Now set r to be 100 times its original value and s to be zero ($r = 0.2$, $s = 0$), and answer the first question again. What kind of market does this represent?

Part 2: Influentials and imitators

In a real marketplace, some adopters are more influential than others. Therefore, to be more realistic, we will now partition the entire population into two groups called *influentials* and *imitators* [65]. Influentials are only influenced by other influentials, while imitators can be influenced by either group. The numbers of influentials and imitators in the population are N_A and N_B , respectively, so the total population size is $N = N_A + N_B$. We will let $A(t)$ now represent the fraction of the influential

population that has adopted the product at time t and let $B(t)$ represent the fraction of the imitator population that has adopted the product at time t .

The adoption rate of the influentials follows the same difference equation as before, except that we will denote the adoption rate and social contagion for the influentials with r_A and s_A .

$$A(t) = A(t - \Delta t) + r_A \cdot (1 - A(t - \Delta t)) \cdot \Delta t + s_A \cdot A(t - \Delta t) \cdot (1 - A(t - \Delta t)) \cdot \Delta t$$

The adoption rate of the imitators will be different because they value the opinions of both influentials and other imitators. Let r_B and s_B represent the adoption rate and social contagion for the imitators. Another parameter, w (between 0 and 1), will indicate how much the imitators value the opinions of the influentials over the other imitators. At the extremes, $w = 1$ means that the imitators are influenced heavily by influentials and not at all by other imitators. On the other hand, $w = 0$ means that they are not at all influenced by influentials, but are influenced by imitators. We will break the difference equation for $B(t)$ into three parts.

1. First, there is a constant rate of adoptions from among the imitators that have not yet adopted, just like the first part of the difference equation for $A(t)$:

$$r_B \cdot (1 - B(t - \Delta t)) \cdot \Delta t$$

2. Second, there is a fraction of the imitators who have not yet adopted who will be influenced to adopt, through social contagion, by influential adopters.

$$w \cdot s_B \cdot \underbrace{A(t - \Delta t)}_{\substack{\text{fraction of} \\ \text{influentials who} \\ \text{have adopted}}} \cdot \underbrace{(1 - B(t - \Delta t))}_{\substack{\text{fraction of} \\ \text{imitators who} \\ \text{have not adopted}}} \cdot \Delta t.$$

Recall from above that w is the extent to which imitators are more likely to be influenced by influentials than other imitators.

3. Third, there is a fraction of the imitators who have not yet adopted who will be influenced to adopt, through social contagion, by other imitators who have already adopted.

$$(1 - w) \cdot s_B \cdot \underbrace{B(t - \Delta t)}_{\substack{\text{fraction of} \\ \text{imitators who} \\ \text{have adopted}}} \cdot \underbrace{(1 - B(t - \Delta t))}_{\substack{\text{fraction of} \\ \text{imitators who} \\ \text{have not adopted}}} \cdot \Delta t.$$

The term $1 - w$ is the extent to which imitators are likely to be influenced by other imitators, compared to influentials.

Putting these three parts together, we have the difference equation modeling the growth of the fraction of imitators who adopt the product.

$$\begin{aligned} B(t) = & B(t - \Delta t) + r_B \cdot (1 - B(t - \Delta t)) \cdot \Delta t \\ & + w \cdot s_B \cdot A(t - \Delta t) \cdot (1 - B(t - \Delta t)) \cdot \Delta t \\ & + (1 - w) \cdot s_B \cdot B(t - \Delta t) \cdot (1 - B(t - \Delta t)) \cdot \Delta t \end{aligned}$$

Now write a function

```
productDiffusion2(inSize, imSize, rIn, sIn, rIm, sIm, weight, weeks, dt)
```

that implements this product diffusion model with influentials and imitators. The parameters are similar to the previous function (but their names have been shortened). The first two parameters are the sizes of the influential and imitator populations, respectively. The third and fourth parameters are the adoption rate (r_A) and social contagion (s_A) for the influentials, respectively. The fifth and sixth parameters are the same values (r_B and s_B) for the imitators. The seventh parameter **weight** is the value of w . Your function should produce two plots. In the first, plot the new adoptions for each group, and the total rate of new adoptions, over time (as before, normalized by dividing by **dt**). In the second, plot the total adoptions for each group, and the total adoptions for both groups together, over time. Unlike in the previous function, plot the *numbers* of adopters in each group rather than the fractions of adopters, so that the different sizes of each population are taken into account. (To do this, just multiply the fraction by the total size of the appropriate population.)

Write a program that uses your function to simulate the same product launch as before, except now there are 600 influentials and 400 imitators in a total population of 1000. The adoption rate and social contagion for the influentials are the same as before ($r_A = 0.002$ and $s_A = 1.03$), but these values are $r_B = 0$ and $s_B = 0.8$ for the imitators. Use a value of $w = 0.6$, meaning that the imitators value the opinions of the influentials over other imitators.

Question 4.3.4 Describe the picture and explain the pattern of new adoptions and the resulting pattern of total adoptions. Point out any patterns that you find interesting.

Question 4.3.5 Now set $w = 0.01$ and rerun the simulation. Describe the new picture and explain how and why this changes the results.

***Project 4.4 Wolves and moose**

For this project, we assume that you have read Section 4.4.

Predator-prey models are commonly used in biology because they can model a wide range of ecological relationships, e.g., wolves and moose, koala and eucalyptus, or humans and tuna. In the simplest incarnation, the livelihood of a population of predators is dependent solely on the availability of a population of prey. The population of prey, in turn, is kept in check by the predators.

In the 1920's, Alfred Lotka and Vito Volterra independently introduced the now-famous Lotka-Volterra equations to model predator-prey relationships. The model consists of a pair of related differential equations that describe the sizes of the two populations over time. We will approximate the differential equations with discrete difference equations. Let's assume that the predators are wolves and the prey are moose. We will represent the sizes of the moose and wolf populations at the end of month t with $M(t)$ and $W(t)$, respectively. The difference equations describing the populations of wolves and moose are:

$$M(t) = M(t - \Delta t) + b_M M(t - \Delta t) \Delta t - d_M W(t - \Delta t) M(t - \Delta t) \Delta t$$

$$W(t) = W(t - \Delta t) + b_W W(t - \Delta t) M(t - \Delta t) \Delta t - d_W W(t - \Delta t) \Delta t$$

where

- b_M is the moose birth rate (per month)
- d_M is the moose death rate, or the rate at which a wolf kills a moose that it encounters (per month)
- b_W is the wolf birth rate, or the moose death rate \times how efficiently an eaten moose produces a new wolf (per month)
- d_W is the wolf death rate (per month)

Let's look at these equations more closely. In the first equation, the term $b_M M(t - \Delta t) \Delta t$ represents the net number of moose births in the last time step, in the absence of wolves, and the term $d_M W(t - \Delta t) M(t - \Delta t) \Delta t$ represents the number of moose deaths in the last time step. Notice that this term is dependent on both the number of wolves and the number of moose: $W(t - \Delta t) M(t - \Delta t)$ is the number of possible wolf-moose encounters and $d_M \Delta t$ is the rate at which a wolf kills a moose that it encounters in a time step of length Δt .

In the second equation, the term $b_W W(t - \Delta t) M(t - \Delta t) \Delta t$ represents the number of wolf births per month. Notice that this number is also proportional to both the number of wolves and the number of moose, the idea being that wolves will give birth to more offspring when food is plentiful. As described above, b_W is actually based on two quantities, the moose death rate (since wolves have to eat moose to thrive and have offspring) and how efficiently a wolf can use the energy gained by eating a moose to give birth to a new wolf. The term $d_W W(t - \Delta t) \Delta t$ represents the net number of wolf deaths per month in the absence of moose.

In this project, you will write a program that uses these difference equations to model the dynamic sizes of a population of wolves and a population of moose over time. There are three parts to the project. In the first part, you will use the Lotka-Volterra model to simulate a baseline scenario. In the second part, you will model the effects that hunting the wolves have on the populations. And, in the third part, you will create a more realistic simulation in which the sizes of the populations are limited by the natural resources available in the area.

Part 1: Implement the Lotka-Volterra model

Write a function in Python

```
PP(preypop, predpop, dt, months)
```

that simulates this predator prey model using the difference equations above. The parameters `preypop` and `predpop` are the initial sizes of the prey and predator populations ($M(0)$ and $W(0)$), respectively, `dt` (Δt) is the time interval used in the simulation, and `months` is the number of months (maximum value of t) for which to run the simulation. To cut back on the number of parameters, you can assign constant birth and death rates to local variables inside your function. Start by trying

```
birthRateMoose = 0.5      #  $b_M$ 
deathRateMoose = 0.02     #  $d_M$ 
birthRateWolves = 0.005   #  $b_W = d_M \times \text{efficiency } 0.25$ 
deathRateWolves = 0.75    #  $d_W$ 
```

Your function should plot, using `matplotlib`, the sizes of the wolf and moose populations over time, as the simulation progresses. Write a program that calls your `PP` function to simulate 500 moose and 25 wolves for 5 years with `dt = 0.01`.

Question 4.4.1 *What happens to the sizes of the populations over time? Why do these changes occur?*

Part 2: Here come the hunters!

Now suppose the wolves begin to threaten local ranchers' livestock (in Wyoming, for example) and the ranchers begin killing the wolves. Simulate this effect by increasing the wolf death rate d_W .

Question 4.4.2 *What is the effect on the moose population?*

Question 4.4.3 *What would the wolf death rate need to be for the wolf population to die out within five years? Note that the death rate can exceed 1. Try increasing the value of d_W slowly and watch what happens. If it seems like you can never kill all the wolves, read on.*

Killing off the wolves appears to be impossible because the equations you are using will never let the value reach zero. (Why?) To compensate, we can set either population to zero when it falls below some threshold, say 1.0. (After all, you can't really have a fraction of a wolf.) To do this, insert the following statements into the body of your `for` loop after you increment the predator and prey populations, and try answering the previous question again.

```

if preyPop < 1.0:
    preyPop = 0.0
if predPop < 1.0:
    predPop = 0.0

```

(Replace `preyPop` and `predPop` with the names you use for the current sizes of the populations.) As we will see shortly, the first two statements will assign 0 to `preyPop` if it is less than 1. The second two statements do the same for `predPop`.

Part 3: Modeling constrained growth

In the simulation so far, we have assumed that a population can grow without bound. For example, if the wolf population died out, the moose population would grow exponentially. In reality, an ecosystem can only support a limited size population due to constraints on space, food, etc. This limit is known as a *carrying capacity*. We can model the moose carrying capacity in a simple way by decreasing the moose birth rate proportionally to the size of the moose population. Specifically, let MCC represent the moose population carrying capacity, which is the maximum number of moose the ecosystem can support. Then change the moose birth term $b_M M(t - \Delta t)\Delta t$ to

$$b_M (1 - M(t - \Delta t)/MCC) M(t - \Delta t)\Delta t.$$

Notice that now, as the moose population size approaches the carrying capacity, the birth rate slows.

Question 4.4.4 *Why does this change cause the moose birth rate to slow as the size of the moose population approaches the carrying capacity?*

Implement this change to your simulation, setting the moose carrying capacity to 750, and run it again with the original birth and death rates, with 500 moose and 25 wolves, for 10 years.

Question 4.4.5 *How does the result differ from your previous run? What does the result demonstrate? Does the moose population reach its carrying capacity of 750? If not, what birth and/or death rate parameters would need to change to allow this to happen?*

Question 4.4.6 *Reinstate the original birth and death rates, and introduce hunting again; now what would the wolf death rate need to be for the wolf population to die out within five years?*