

MATH 367 Assignment

Ian McKenzie, Scott Nerbas

2017-04-01

1 Exercise 1

Provide justifications for the first four assumptions.

1.1

The first assumption is probably reasonable for small populations. Obviously it cannot hold in real life for arbitrarily large populations, as resources such as rabbit food are scarce in nature.

1.2

The rabbits being eaten at a rate proportional to the size of both rabbit and wolf populations seems reasonable. Immortal rabbits seems a bit odd at first, but when the wolf population is large enough compared to the rabbit population it seems likely that the only source of rabbit deaths would be being eaten by wolves.

1.3

Rabbits being the only source of food for wolves seems reasonable considering the nature of the small island and assuming that there are only 2 species on the entire island.

1.4

Mortal wolves seems to approximate real life very well, however a more sophisticated model would likely take into account disease or other predators.

2 Exercise 2

Suppose $x(t) = 0$ for all t . What is $y(t)$? Suppose $y(t) = 0$ for all t . What is $x(t)$? Are these answers reasonable?

2.1

When $x(t) = 0$,

$$\begin{aligned}y'(t) &= -d y(t) \\ \frac{y'(t)}{y(t)} &= -d \\ \int \frac{y'(t)}{y(t)} dt &= \int -d dt \\ &\vdots \\ y(t) &= k e^{-dt}\end{aligned}$$

where k is some constant.

2.2

When $y(t) = 0$,

$$\begin{aligned} x'(t) &= a x(t) \\ \frac{x'(t)}{x(t)} &= a \\ \int \frac{x'(t)}{x(t)} dt &= \int a dt \\ &\vdots \\ x(t) &= k e^{at} \end{aligned}$$

where k is some constant.

3 Exercise 3

3.1

If $x(t) \geq 0$ and $x'(t) = 0$ then $0 = x(t) (a - by(t))$. This means that $a - by(t)$ must be 0, so:

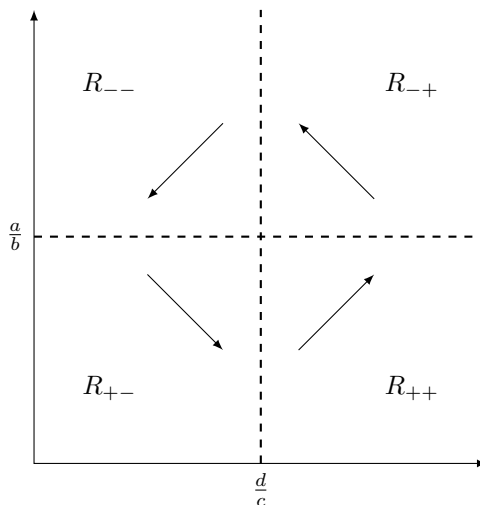
$$y(t) = \frac{a}{b}$$

If $y(t) \geq 0$ and $y'(t) = 0$ then $0 = y(t) (cx(t) - d)$. This means that $cx(t) - d$ must be 0, so:

$$x(t) = \frac{d}{c}$$

3.2

Parametric population trajectory curves are indicated by arrows.



4 Excercise 4

By first order approximation,

$$y(t_1) = y(t_0 + \Delta t) \approx y(t_0) + y'(t_0)\Delta t = y_0 + y'(t_0)\Delta t$$

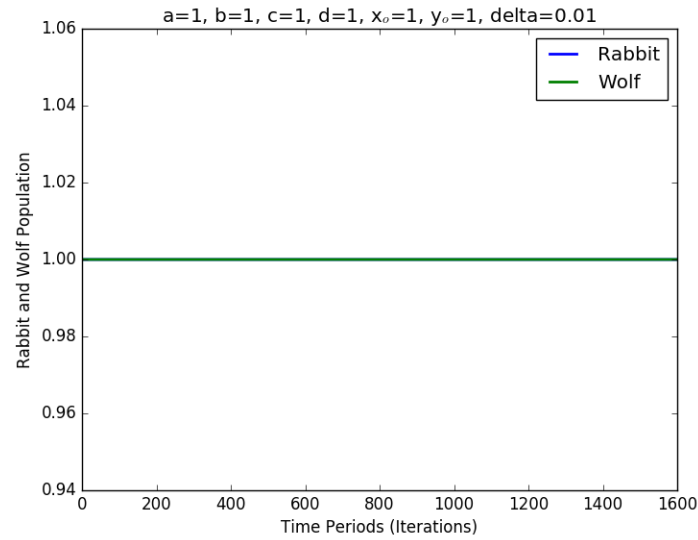
and,

$$y'(t_0) = cx(t_0)y(t_0) - dy(t_0) = cx_0y_0 - dy_0.$$

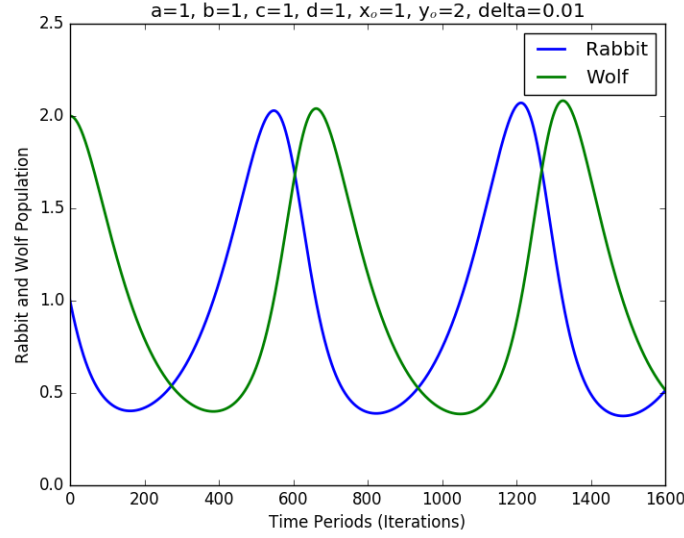
Therefore,

$$y(t_1) = y(t_0 + \Delta t) \approx y_0 + (cx_0y_0 - dy_0)\Delta t.$$

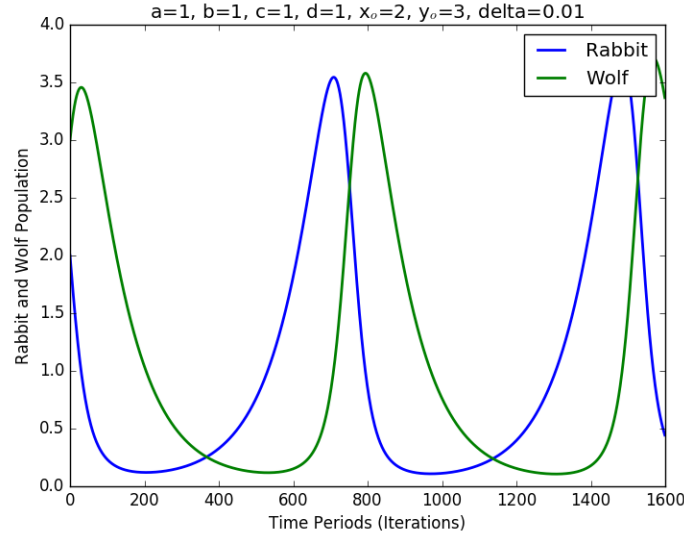
5 Exercise 5



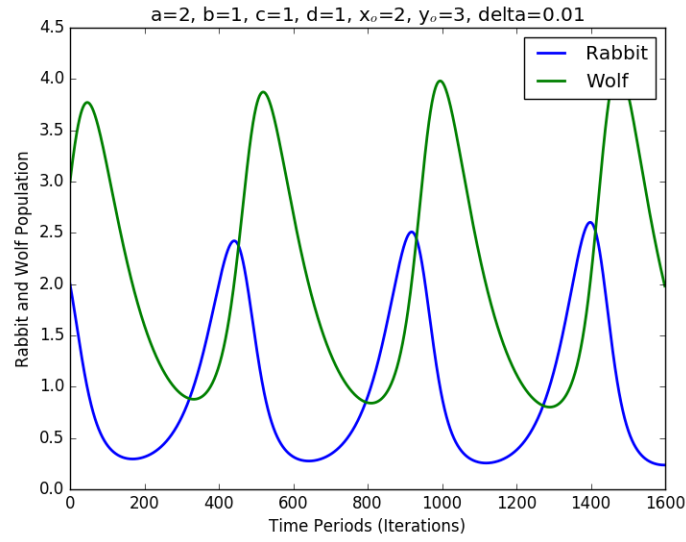
In this simulation, we used the constants a, b, c, d all equal to 1. The starting population values are both 1. The populations do not grow or decline from their initial values.



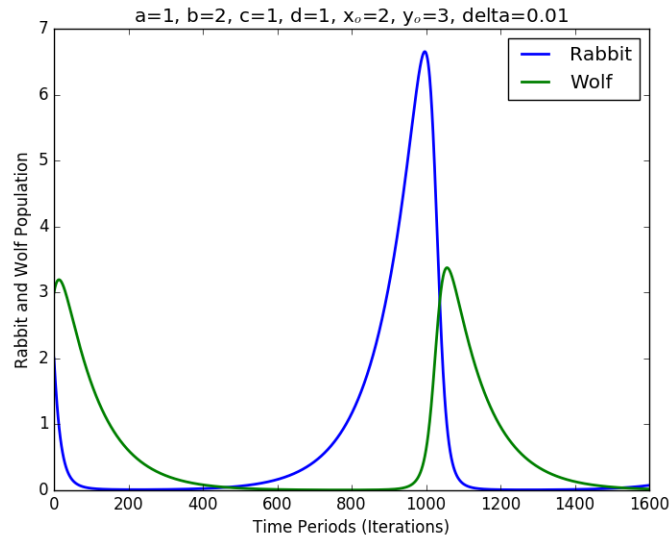
In this simulation, we used the constants a, b, c, d all equal to 1. The starting population values are $x_0 = 1$ for rabbits and $y_0 = 2$ for wolves. The rabbit and the wolf functions appear to have the same maximums and minimum values in this case.



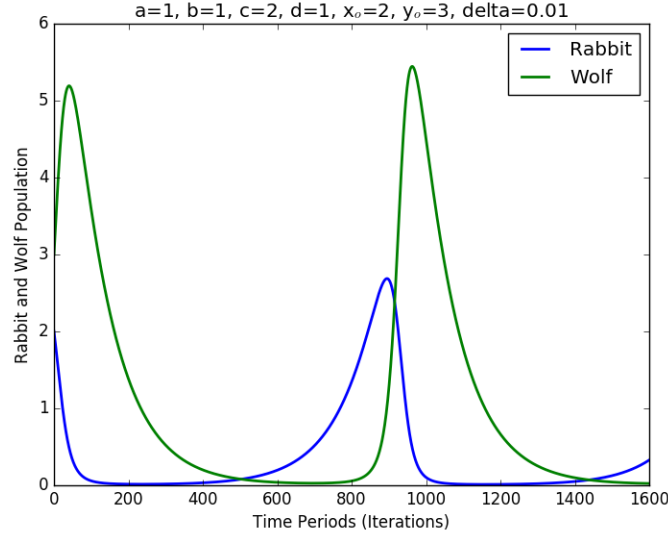
In this simulation, we used the constants a, b, c, d all equal to 1. The starting population values are $x_0 = 2$ for rabbits and $y_0 = 3$ for wolves. The rabbit function appears to have a higher maximum than the wolf function, and the wolf function appears to have the same minimum value as the rabbit function. Furthermore, the functions appear to be mirror images of each other.



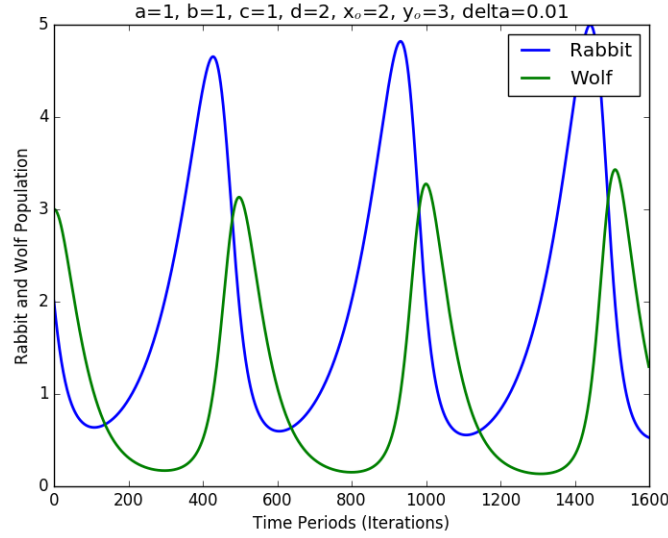
In this simulation, we used the constants $a = 2, b = 1, c = 1, d = 1$ and starting population values of $x_0 = 2$ and $y_0 = 3$. The wolf function appears to have a higher maximum and minimum than the rabbit function.



In this simulation, we used the constants $a = 1, b = 2, c = 1, d = 1$ and starting population values of $x_0 = 2$ and $y_0 = 3$. The rabbit population dramatically increases in size before collapsing as the wolf population increases.



In this simulation, we used the constants $a = 1, b = 1, c = 2, d = 1$ and starting population values of $x_0 = 2$ and $y_0 = 3$. As the wolf population suddenly decreases, the rabbit population rapidly rises.



In this simulation, we used the constants $a = 1, b = 1, c = 1, d = 2$. The starting population values are $x_0 = 2$ and $y_0 = 3$. The rabbit function appears to have a higher maximum and minimum than the wolf function.

6 Exercise 6

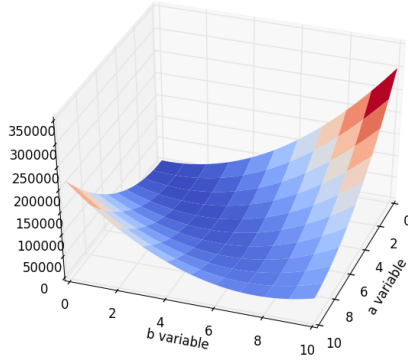
We used initial population values of $x_0 = 2, y_0 = 1$. We use the gradient descent algorithm for minimization.

6.1 (a, b)

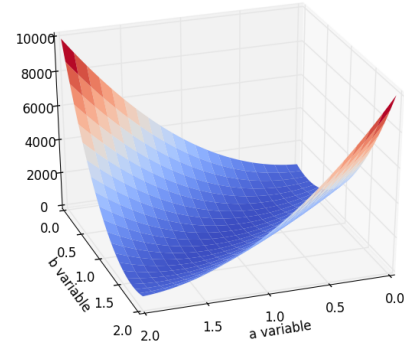
The gradient of the (a, b) error function is:

$$\nabla E_{a,b} = \left(\sum_{k=1}^n -2x_k \left(\frac{x_k - x_{k-1}}{t_k - t_{k-1}} - (ax_k - bx_k y_k) \right), \sum_{k=1}^n 2x_k y_k \left(\frac{x_k - x_{k-1}}{t_k - t_{k-1}} - (ax_k - bx_k y_k) \right) \right)$$

To avoid local minimums, we first plot the error function so we can start our gradient descent near the absolute minimum.

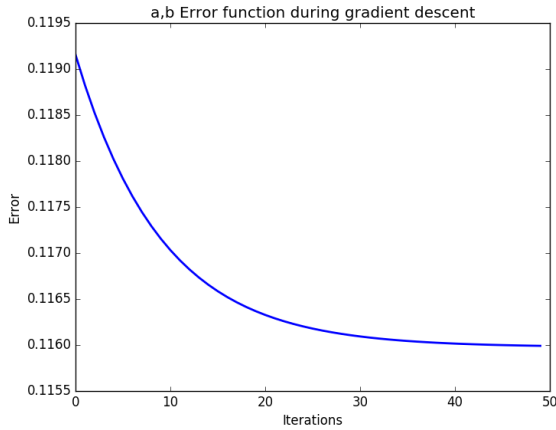


Large plot.

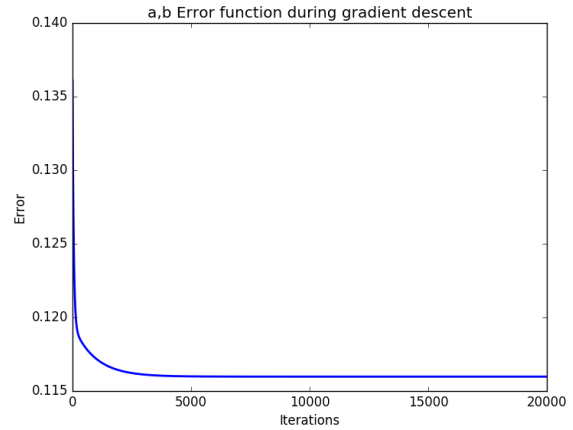


Near origin.

From this it appears that the minimum is near the origin, so we produce another plot with less range to illuminate the area near the origin. From this second plot it seems that the minimum occurs near the point $a = b = 1$, so we begin our gradient descent at that point.



50 iterations, larger Δ .



Actual convergence. 20,000 iterations, smaller Δ .

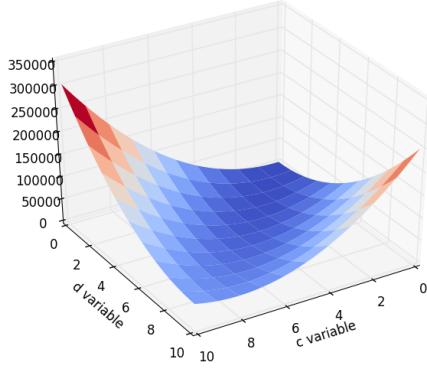
To illustrate how our gradient descent converges, we show the error over 50 iterations with a step size of 0.0001. To find a more accurate minimum, we computed 20,000 iterations with a step size of 0.000001. This resulted in the minimum of 0.11598 which occurs at $a = 1.00398, b = 1.00084$. From the actual convergence plot, we can see that 20,000 iterations is probably excessive, but at least we know that the minimum is accurate.

6.2 (c, d)

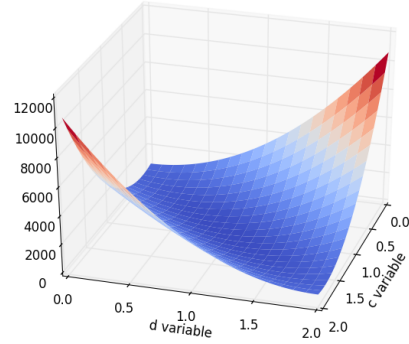
The gradient of the (c, d) error function is:

$$\nabla E_{c,d} = \left(\sum_{k=1}^n -2x_k y_k \left(\frac{y_k - y_{k-1}}{t_k - t_{k-1}} - (cx_k y_k - dy_k) \right), \sum_{k=1}^n 2y_k \left(\frac{y_k - y_{k-1}}{t_k - t_{k-1}} - (cx_k y_k - dy_k) \right) \right)$$

Again, to avoid local minimums, we first plot the error functions so we can start our gradient descent near the absolute minimum.

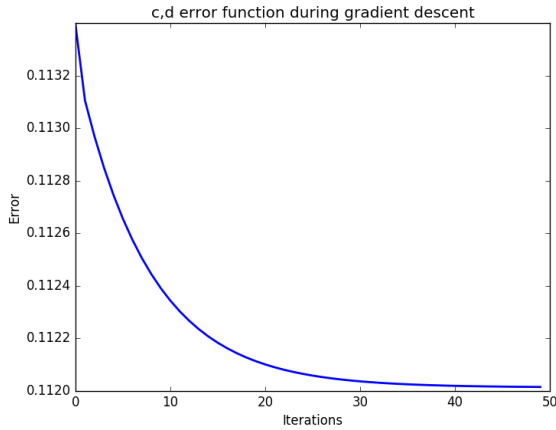


Large plot.

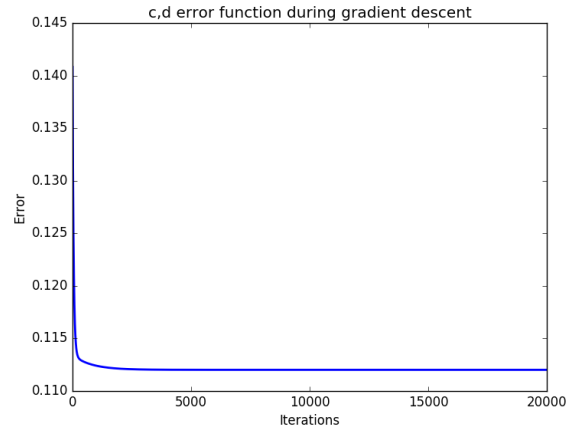


Near origin.

From this it appears that the minimum occurs near the origin, so we produced another plot with less range to illuminate the area near the origin. From this second plot it seems that the minimum occurs near the point $c = d = 1$, so we begin our gradient descent at that point.



50 iterations, larger Δ .



Actual convergence. 20,000 iterations, smaller Δ .

To illustrate how our gradient descent converges, we show the error over 50 iterations with a step size of 0.0001. To find a more accurate minimum, we completed 20,000 iterations with a step size of 0.000001. This resulted in the minimum of 0.11201 which occurs at $c = 1.00034, d = 0.99691$. From the plot we can see that 20,000 iterations is probably excessive, but at least we know that the minimum is accurate.