Predator-Prey-Scavenger model

Darby Vaughn

May 3, 2018

Abstract

The Lotka-Volterra equations, commonly called the predator-prey equations, are used to model populations dynamics between two or more species. These equations are used to model the interaction of two species; we intend to expand this to take into account a third species, whose role will be a scavenger. To model our equations, we will use a 4-stage Runge-Kutta method. We are interested in finding stability in our model, along with analyzing the Runge-Kutta method we intend to use.

1 Introduction

We intend to apply the predator-prey model to a specific example using a numerical method to approximate the result. First, we will analyze a numerical method for convergence, order of convergence, exactness, and stability. Next, we will introduce our initial value problem, and the variables along with their meaning. Finally, we will use some specific data to test our numerical method, and draw conclusions based on the results.

2 Numerical Method Description

The numerical method we will be using to approximate our IVP is a 4-stage explicit R-K method. R-K methods are a family of implicit or explicit methods used to approximate the solutions to ordinary differential equations. The general 4-stage implicit R-K method is described as follows:

$$
\begin{aligned}
\xi_1 &= y_n \\
\xi_2 &= y_n + h * a_{21} * f(t_n + h * x_1, \xi_1) \\
\xi_3 &= y_n + h * [a_{31} * f(t_n + h * x_1, \xi_1) + a_{32} * f(t_n + h * x_2, \xi_2)] \\
\xi_4 &= y_n + h * [a_{41} * f(t_n + h * x_1, \xi_1) + a_{42} * f(t_n + h * x_2, \xi_2)] + a_{43} * f(t_n + h * x_3, \xi_3)] \\
y_{n+1} &= y_n + h * [b_1 * f(t_n + h * x_1, \xi_1) + b_2 * f(t_n + h * x_2, \xi_2)] + b_3 * f(t_n + h * x_3, \xi_3) + b_4 * f(t_n + h * x_4, \xi_4)]
\end{aligned}
$$

where a_{ij} correspond to the entries in an R-K matrix, x_i's correspond to the R-K nodes, and b_i's correspond to R-K weights, and h is the step-size that we will use to determine the points at which we wish to evaluate the R-K method. A smaller step-size will give us a more accurate approximation, as the distance in between each approximation will be smaller. In our particular case we will be using the values given by the R-K tableau shown below.

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}
$$

This will give us the following R-K method:

$$\begin{aligned}
\xi_1 &= y_n \\
\xi_2 &= y_n + h/2 * f(t_n, \xi_1) \\
\xi_3 &= y_n + h/2 * f(t_n + h/2, \xi_2) \\
\xi_4 &= y_n + h * f(t_n + h/2, \xi_3) \\
y_{n+1} &= y_n + h * [1/6 * f(t_n, \xi_1) + 1/3 * f(t_n + h/2, \xi_2)] + 1/3 * f(t_n + h/2, \xi_3) + 1/6 * f(t_n + h, \xi_4)]
\end{aligned}$$

We know that this R-K method will converge, so long as it satisfies the two following conditions:

1. its order p≥1.
2. the polynomial $\rho(\omega)$ obeys the root condition.

To show that the order of convergence of this 4-stage R-K method is 4, we will apply the following theorem:

Theorem 1 *The order of an S-step method is p* $\iff \rho(\omega) - \sigma(\omega)\ln(\omega) = \mathcal{O}\left|\omega - 1\right|^{p+1}$ *as* $\omega \to 1$

To do this, we first need to define what $\rho(\omega)$ and $\sigma(\omega)$ are

$$\rho(\omega) = \sum_{m=0}^{s} a_m \omega^m$$
$$\sigma(\omega) = \sum_{m=0}^{s} b_m \omega^m$$

where a_m is the value of the constant in front of y_{n+m}, and b_m is the value of the constant in front of $f(t_{n+m}, y_{n+m})$. In our particular case, we have that

$$\rho(\omega) = \omega - 1$$
$$\sigma(\omega) = \frac{1}{6}\omega^3 + \frac{1}{3}\omega^2 + \frac{1}{3}\omega + \frac{1}{6}$$

Applying these particular equations to our theorem, we will get that

$$\rho(\omega) - \sigma(\omega)\ln(\omega) = \mathcal{O}\left|\omega - 1\right|^5$$

So our R-K method has order p=4, and thus fulfills the first requirement. To see that our R-K method fulfills the second requirement, we notice that $\rho(\omega)$ has one root of magnitude one, and thus $\rho(\omega)$ fulfills the root condition. Hence, we know that our R-K method in fact does converge, and that its order of convergence is p=4.This also means that our method will be exact for quartic polynomials.

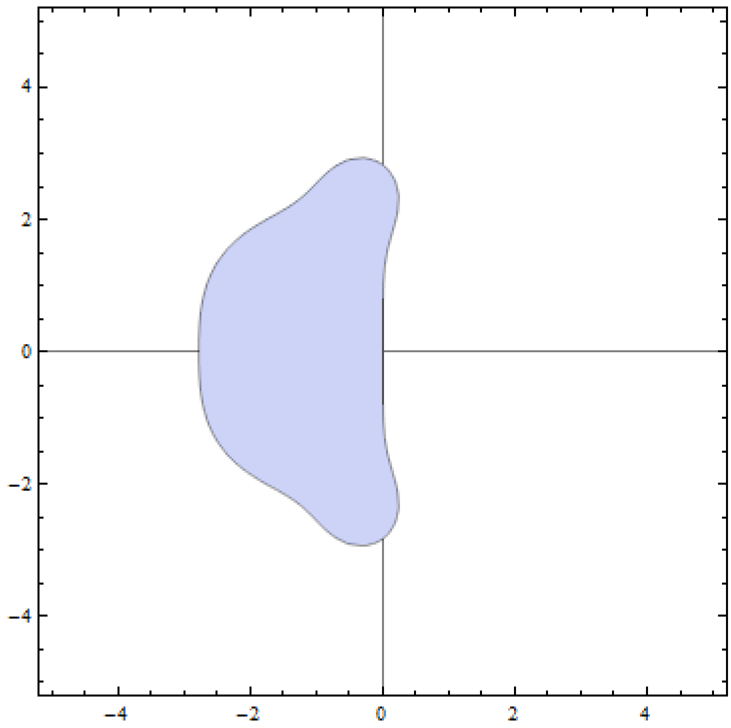
Lastly, we wish the check and see if our R-K method is A-stable or not, and if it is not, then what the stability domain is for this R-K method. To find the stability domain, we will use the following inequality:

$$\left|1 + z * \vec{b}^T(\vec{I} - z\vec{A})^{-1}\vec{1}\right| < 1$$

Where z is a variable in the complex plane, $\vec{A}$ refers to the matrix and $\vec{b}$ refers to the weights given by the R-K tableau, and $\vec{1}$ is a vector of ones with the same number of rows that $\vec{b}$ has. For our R-K method, we get

$$1 + z * \vec{b^T}(\vec{I} - z\vec{A})^{-1}\vec{1} = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4$$

Graphing the inequality $\left|1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4\right| < 1$ in the complex plane, we get the following picture.



This is our stability domain, and we will choose our step-size, h, respectively to ensure that our numerical method is within this region.

3 IVP (Initial Value Problem)

For our predator-prey method, we will use the initial value problem given by

$$\begin{cases} x'(t) = a * x - b * x * y \\ y'(t) = c * x * y - d * y \\ x(t) = \alpha \\ y(t) = \beta \end{cases}$$

This will model the population of the two species in the environment. α is the initial number of species "x" living in the environment at the start of the simulation, and β is the initial number of species "y" living in the environment at the start of the simulation. a and d are positive values describing the growth rate of the two populations. The growth rate of species "x" is positive, and therefore in the absence of species "y", species "x" will grow exponentially. However, the growth rate of species "y" is negative, so without species "x", species "y" will die out. b and c are positive values describing the interactions between the two species. The interaction in x'(t) is negative, which means that the interaction between species "x" and "y" is harmful for species "x". Similarly, the interaction in y'(t) is positive, and therefore the interaction is beneficial for species "y". This combined information tells us the species "x" is the prey and species "y" is the predator in this simulation. This model makes some assumptions that are not necessarily true in nature. It assumes that the prey will have ampule food in the environment, and that the food supply for the predator is completely based on the size of the population of the prey. All this taken into account, the model is still considered to be a good estimator for the interaction of two species.

To convert this information to a code suitable for Matlab, we will convert x'(t) and y'(t) into a matrix with two rows, and assign x(t) to the first entry of the first row of the matrix, and y(t) to the first entry of the second row of the matrix. We also define x'(t) and y'(t) into a function with defined by a matrix, where the first row is x'(t) and the second row is y'(t). the result gives us the following code in Matlab:

$$F = @(y,t)[a*x - b*x*y; c*x*y - d*y]$$
$$y(1,1) = \alpha$$
$$y(2,1) = \beta$$

We can expand this model to take into account another species interacting with the previous two. A scavenger is an interesting species to take into consideration, as it will receive a positive benefit from both the predator and the prey in this model, but will receive no negative interaction from the predator or the prey. It is fairly obvious that a scavenger would have no negative interaction from the prey, as the prey would not hunt it, or compete for food with it. Under this model, we will also assume that the predator does not hunt the scavenger, for instance it is unlikely that a lynx would hunt a turkey vulture, as the vulture could just fly away, and would be too difficult to catch. However, the scavenger would receive a benefit from both the predator and the prey, as it would feed off the carcases of both groups. To model this interaction, we will use the following initial value problem, with all parameters being positive.

$$\begin{cases} x'(t) = a*x - b*x*y \\ y'(t) = c*x*y - d*y \\ z'(t) = e*x*z + f*y*z - g*z \\ x(t) = \alpha \\ y(t) = \beta \\ z(t) = \gamma \end{cases}$$

Notice that x'(t) and y'(t) remain the same, meaning that the scavenger does not give any benefit or disadvantage to the predator or the prey. Also, the values for the interaction between the scavenger and both the predator and the prey are positive, so the scavenger receives a benefit from both of them. However, g is given a negative value implying that in the absence of both the predator and the prey, the scavenger will have no food to eat, and die off.

Converting this initial value problem to Matlab code is fairly similar to the previous code we have shown.

$$F = @(y,t)[a*x - b*x*y; c*x*y - d*y; e*x*z + f*y*z - g*z]$$
$$y(1,1) = \alpha$$
$$y(2,1) = \beta$$
$$y(3,1) = \gamma$$

4 Numerical results

What we will first do is make sure that our Matlab code is running properly for our 4-stage R-K method. To do this, we will model an initial value problem for which we know the exact solution. We will also decrease the step size by $1/2$ for each test, as to see if the R-K method is converging properly. The ratio of the error between each change in step size should be around 2^4, as the method has order of convergence p=4. We will use the following initial value problem, as we know the exact solution:

$$\begin{cases} x'(t) = .4*x \\ y'(t) = -.3*y \\ x(t) = 1 \\ y(t) = 1 \end{cases}$$

Using Matlab to solve this Initial value problem, we get the following ratios of errors:

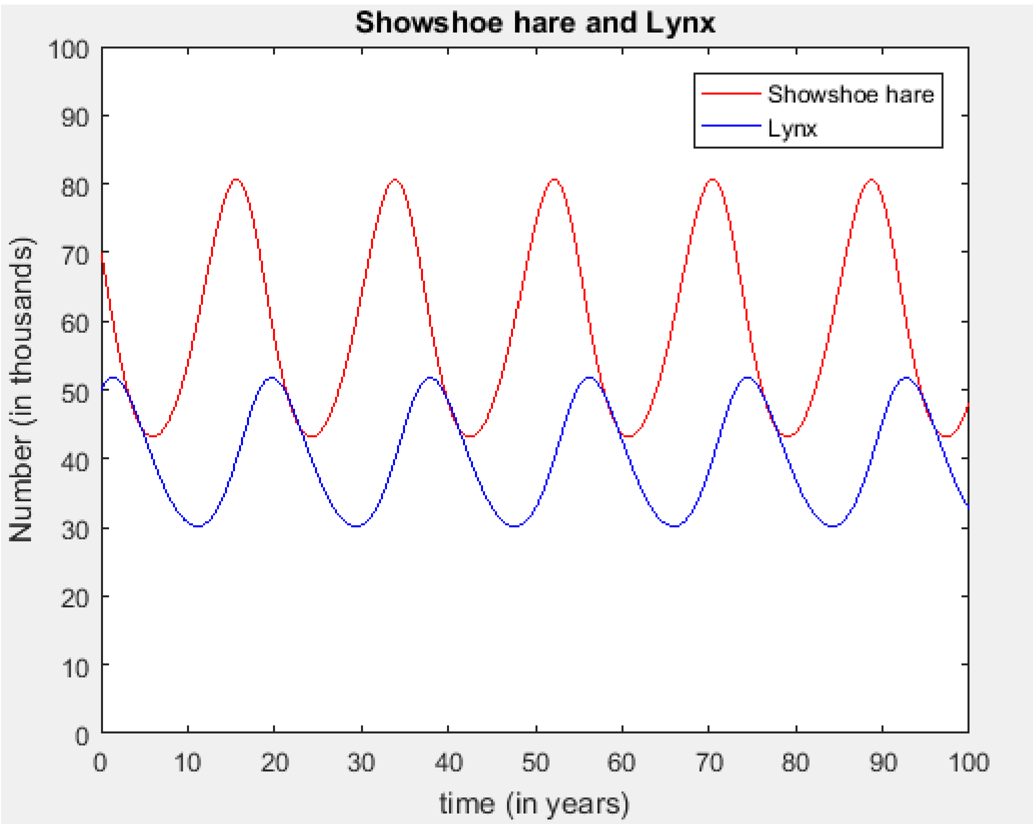
```
error1 =

   13.5568   14.7249   15.3482


error2 =

   18.0676   17.0196   16.5046
```

Error1 is referring to the ratio of the error between the R-K method and $y = e^{.4t}$, and error2 is referring to the ratio of the error R-K method and $y = e^{-.3t}$. We can see that the ratios of the errors are all around 2^4, so our R-K method is behaving properly. This also means that out R-k method is exact for quartic polynomials.

Knowing this, we can move ahead to a specific initial value problem related to the predator-prey method we wish to test. We will be testing the interaction between the Showshoe hare, and the lynx. The Snowshoe hare is the prey, and the lynx is the predator. Both these animals are native to northern America and Canada, which is where the interaction of the two species occurs. To model this interaction, we will use the following initial value problem:

$$\begin{cases} x'(t) = .4x - .1xy \\ y'(t) = .3xy - .005y \\ x(t) = 70 \\ y(t) = 50 \end{cases}$$

We will model this in Matlab using our 4-stage R-K method, and as a reference, we will compare this to ode45 in Matlab to find the ratios of the errors. To find the ratios of the errors, we will compare our 4-stage R-K method with ode45 over several different step sizes. However, since the graphs look almost identical, we will only show the most accurate graph, which is the one with the smallest step size.

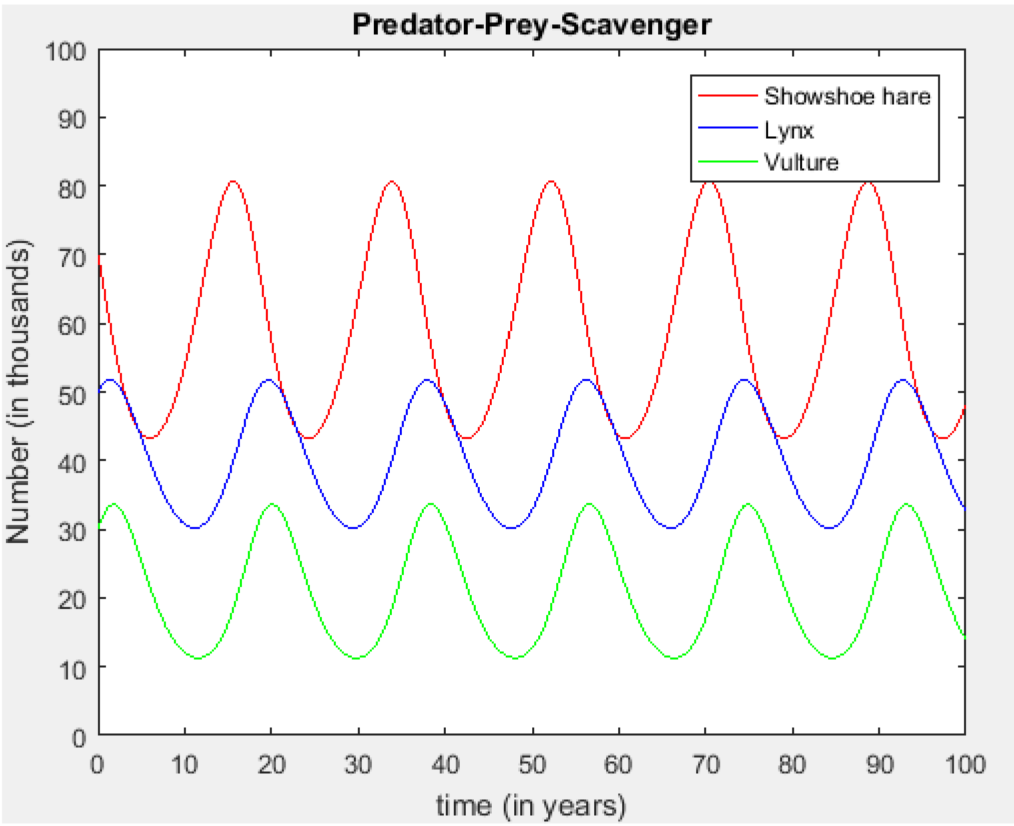


As you can see, the graph shows stability, as neither species dies out over time, which is the desired result. Interestingly, the ratios of the errors of convergence is not near 2^4. This is most likely due to the fact that ode45 is not an exact solution, but an approximation as well. However, we do know that our R-K method does have order of convergence p=4, as we have proven before.

As mentioned before, we will test our model with a scavenger involved also. To do this, we will use the following initial value problem:

$$\begin{cases} x'(t) = .4x - .01xy \\ y'(t) = .3xy - .005y \\ z'(t) = .01xz + .0025yz - .7z \\ x(t) = 70 \\ y(t) = 50 \\ z(t) = 30 \end{cases}$$

This gives us the following graph:

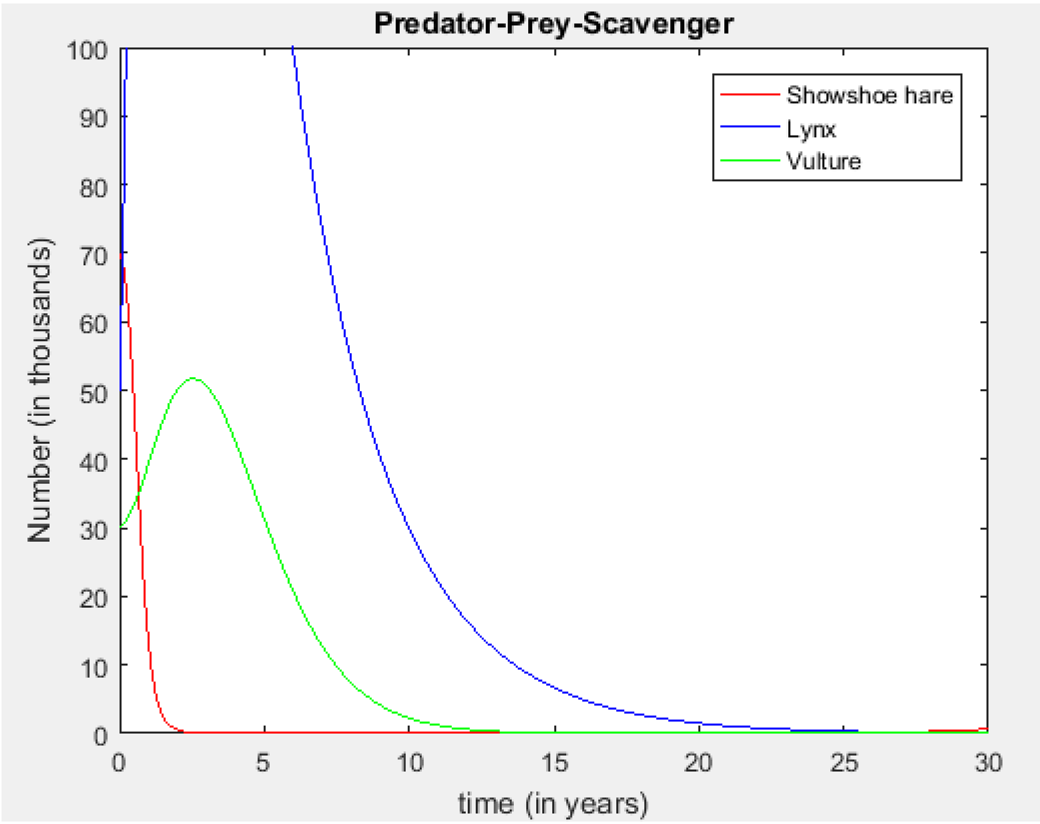


Again, this shows stability, as neither species dies out over time. The predator and prey remain the same in this model, and the scavenger is a vulture, or more specifically a turkey vulture. Turkey vultures are native to North America, and have a range of northern America and Canada in which they overlap with both Showshoe hares and Lynx.

As these previous two graphs have been stable, it is natural to ask if any parameters will provide a stable solution, where no species dies out over time. This of course is not true, the initial value problem given below demonstrates a model where all three species will die out.

$$\begin{cases} x'(t) = .4x - .01xy \\ y'(t) = .3xy - .05y \\ z'(t) = .01xz + .0025yz - .7z \\ x(t) = 70 \\ y(t) = 50 \\ z(t) = 30 \end{cases}$$

As you can see, the only change is the death rate of the predator. This makes sense, as the longer a predator is able to live, the more prey it would have to eat to survive. This leads to over-hunting of the prey, and eventually its population dying out. This gives us the following graph:



Naturally, starting out with too many or too few of a certain species will also lead to the extinction of at least one of them. In our three species model, the only the lynx and vulture can go extinct, without causing all species to go extinct. This is due to the fact that the showshoe hare is the main source of food for the lynx and without the showshoe hare, both the lynx and the vulture will die out.

References

http://math.bd.psu.edu/faculty/jprevite/kathleen/omni2/omnivores3.pdf
http://faculty.sfasu.edu/judsontw/ode/html/firstlook01.html