Final Project

Using Multiple Models to Study an Ecosystem

Written by

Prof. Erin N. Bodine

Prepared for

Math 214: Discrete Math Modeling (Fall 2014)

Rhodes College

The class meetings on 12/2, 12/4, and 12/9 are dedicated to working on this project.

Final Project Due Date:
Monday, 12/15 at 8:00 PM

In Lab 11 we developed a simple model for the ferret-skink-rabbit food web. In this final project, you will expand on that project to develop a more sophisticated and accurate ABM. Additionally, you will use data generated from the ABM to build a matrix model of landscape dynamics

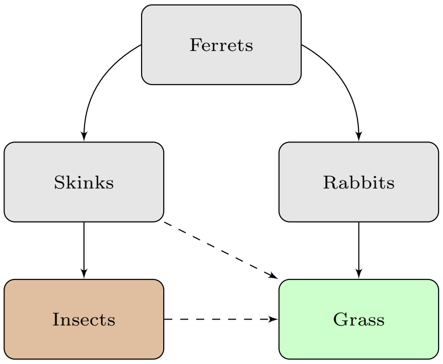

Figure 1: Illustration of ferret-skink-rabbit food web. Solid lines indicate food source; dashed lines indicate habitat use.

In this final project we return to the modified the Rabbits ABM which simulates the more complex ecosystem illustrated in Figure 1. This systems is modeled after the dynamics described in [1]. Skinks are a type of lizard native to New Zealand's dry grasslands, however, after the introduction of European rabbits (*Oryctolagus cuniculus*) many of the native skink species started to decline. The rabbits and the skinks share a common predator, ferrets. The rabbits are the primary food source for the ferrets, and thus when the rabbit population increases, there is a corresponding increase in the ferret population. However, when the rabbit population decreases, the ferret population preys on skinks with increasing frequency, which can have devastating effects on the skink population if the ferret population is still large during a rapid rabbit population decline. Thus, there is a form of competition, known as ***apparent competition***, between the rabbits and the skinks occurring (mediated through the ferret predators).

(a) Endangered Otago skink.

(b) European rabbit.

Figure 2: Prey sources for New Zealand ferrets.

There is however, another layer of complexity. Both the skinks and their food source (a variety of insects), use the rabbits main food source, grass, as their habitat. When the rabbit population increases to an extremely high density, it ravages the grassland habitat of the skinks and their food source. This has a two-fold effect. It reduce the abundance of the skinks food source, and it removes natural cover that would afford some measure of protection from predators like ferrets.

In Lab 11, you simulated the interaction between the ferrets, skinks, rabbits, and grass. Recall that one of the possible outcomes of this model was that the population of skinks could explode. This occurred when the rabbit population died off and there was nothing limiting the food source of the skinks. In this final project, you will add in the insect population, along with many other modification to create a more realistic simulation of the ferrets-skinks-rabbits-grass-insects ecosystem. Additionally, we will test the model under various initial condition to better examine the interdependency of the food-web.

Lastly, note that in previous labs, you were given a set of step-by-step instruction about how to modify existing code to create the desired model. However, in this lab, you will only be given the ODD description of the model, and you must work to construct the model using the skills you have gained in previous labs and examples of code from previous labs. As with previous labs, do not be shy about using the NetLogo Dictionary to look-up how to use a particular primitive, or to search for a primitive which might prove useful to accomplish a particular task.

ODD Description of Model

Recall that ODD stands for Overview, Design Concepts, and Details. The following is a description of the model you will encode into NetLogo that follows the ODD protocol. Note that

- The Process Overview section corresponds to the GO procedure,
- The Initialization section corresponds to the SETUP procedure,
- Each submodel corresponds to an additional procedure you will need to code into the model, and
- The Observation section corresponds to what outputs are recorded in the NetLogo interface.

Overview

Purpose

The purpose of this model is to simulate the ferret-rabbit-skink-insect-grass food-web shown in Figure 1, and to use the model to determine the average fate of the ecosystem under various initial population structures. The biology of the system is describe on the previous page.

Entities, State Variables, Scales

This model contains four different types (or breeds) of agents, and a set of square patches representing the model world. The four types of agents are `ferrets`, `rabbits`, `skinks`, and `bugs`. Each type of agent has state variables for their location in the model world, their shape and color representation within the visual representation of the model, their current heading, and the amount of `energy` they currently have (at any particular point within a simulation).

The model world is represented by square grid 41 patches wide and 41 patches tall. The origin of the grid is located in the center of the grid, and thus the x and y coordinates on the grid both fall in the range $[-20, 20]$. Additionally, the model world represents a torus, and thus wraps both vertically and horizontally. The area of one patch corresponds to the amount of grass one rabbit could eat in one day. Lastly, each patch has state variables corresponding to its color and the amount of energy any plant material growing on that patch contains. This latter variable is referred to as `grass-energy`.

Within the model black patch (⬛) represent bare land, each green patch (🟩) represents grass, and each yellow patch (🟨) represents grass with dandelions. The rate at which black bare land grows grass (black to green) and at which grass grows dandelions (green to yellow) is given by the global parameter `grass-grow-rate`. The rate at which grass and grass with dandelion patches revert back to bare land, is computed using several global variables

- `countG` – the number of green patches
- `countY` – the number of yellow patches
- `countGtoB` – the number of green patches that have become black
- `countYtoB` – the number of yellow patches that have become black

- `rateGtoB` – the proportion of green patches that became black
- `rateYtoB` – the proportion of yellow patches that became black

In addition to the global variables there are three global parameters. The parameter `bug-energy` defines the amount of energy contained in each bug agent, `bugs-per-patch` defines the number of bugs that sprout to life on a new grown patch of green grass, and `grass-grow-rate` defines the probability that a bare patch grows grass, and that a grass patch grows dandelions.

Lastly, it should be noted that each time step represents 1 day.

Process Overview

Within a single time step of the model the following sequence of events occurs:

1. The simulation is halted if there are no rabbits, ferrets, or skinks left in the ecosystem.
2. Each patch executes the GROW-GRASS submodel.
3. The values of `countGtoB` and `countYtoB` are initialized to zero, and the values of `countG` and `countY` are set to the current number of green and yellow patches, respectively. Note, this is done prior to any grass (🟩) or grass with dandelions (🟨) patches being eaten or naturally dying, which would cause them to revert to bare land (⬛).
4. All of the agents who are not bugs execute the following:
   - First the agent moves (if necessary) and eats. If the agent is a skink on a patch with bugs, then it remains on that patch and executes the EAT-BUG submodel. Otherwise, the agent executes the MOVE submodel, and then if the agent is a rabbit executes the EAT-GRASS submodel, and if the agent is a ferret executes the EAT-PREY submodel.
   - Next, the agent executes the REPRODUCE submodel. This will allow the agent to reproduce if they have the requisite amount of energy.
   - Lastly, the agent executes the DEATH submodel in which the agent will die if they no longer have enough energy to sustain their activities.
5. Each patch GRASS-DEATH submodel in which patches of grass (🟩) and grass with dandelions (🟨) will "age" by losing some energy. If they lose all of their energy, they revert back to bare land (⬛).
6. The proportions of green and yellow patches that became black are calculated using the formulas

   $$\texttt{rateGtoB} = \frac{\texttt{countGtoB}}{\texttt{countG} + 1} \qquad \text{and} \qquad \texttt{rateYtoB} = \frac{\texttt{countYtoB}}{\texttt{countY} + 1}$$

   Note the "+1" in the denominator of each calculation, this is to avoid dividing by zero in the event that there were initially no grass or grass with dandelions patches.
7. The time step counter (`ticks`) is increased by 1.

Design Concepts

Basic Concepts This model uses many of the same basic design elements as the `Rabbits.nlogo` model explored and expanded on in previous lab projects in this class. In fact, some of the submodels from those previous lab projects will remain unchanged in this model.

Emergence When the population of ferrets and one of its prey can be sustained over time, a spatial pattern may emerge where it appears that the ferrets are “chasing” their prey across the landscape. However, it should be noted that there is nothing within the code that would cause the heading of a ferret to align with that of the prey. This emergent property can be explained through a “survival of the fittest” type argument. The fittest ferrets (the ones able to accumulate enough energy to reproduce) will be the ones who are moving in the direction of their prey. Since the offspring of the ferrets inherit the heading direction of their parent and move forward one patch length, this gives the appearance of a wave of ferrets chasing their prey across the landscape.

Adaptation The only adapation occurs within the skinks when they choose to move only if there is no food available (i.e. bugs) on the patch on which they are currently located.

Objectives The objective of the ferret, rabbit, and skink agents is to maintain enough energy to stay alive. The bug agents have no objectives.

Learning There is no learning in this ABM.

Prediction None of the agents make any predictions in this ABM.

Sensing The ferret, rabbit, and skink agents can sense if they are on a patch with an appropriate food source.

Interaction The ferrets interact with the rabbits and skinks directly by consuming them as prey sources (see the eat-prey submodel for details). The ferrets have no direct interaction with the bugs or the patches. The rabbits interact with the patches by consuming the energy on patches with grass (🟩) or grass with dandelions (🟨). This causes an indirect interaction with the bugs that reside in the grass or dandelions, as the bugs cannot survive on bare land, thus killing any bugs previously occupying that patch. The skinks interact directly with the bugs by consuming them. Additionally, the patches provide protection for the skinks from predation by the ferrets.

Stochasticity The direction in which each of the ferret, rabbit, and skink agents move at each time step (if they move) is randomly chosen to be within 50° of the direction in which they are currently heading. Additionally, if the ferret finds itself on a patch with multiple prey, it will randomly select one of the prey to consume. If the ferret selects a skink on a patch of grass or grass with dandelions, there is only a 20% probability that the ferret will succeed in consuming its prey, otherwise (i.e., if the ferret selects a skink on bare land or a rabbit anywhere) there is an 80% probability that the ferret will succeed in consuming its prey.

Collectives There are no collectives in this ABM.

Observation In addition to the visual display of the agents moving throughout the landscape over time, two graphs are produced with data recorded at each time step. The first is a graph showing the number of each type of agent, and the total number of non-bare land patches (i.e., patches with grass (🟩) or with grass and dandelions (🟨)). The second graph shows how the land scape is changing by plotting the proportion of green and yellow patches which revert back to black patches (i.e., `rateGtoB` and `rateYtoB`) at each time step.

Details

Initialization

The initial number of rabbits, ferrets, and skinks are set using input boxes in the model interface. These initial values can be varied between simulations. The global parameter `bug-energy` which represents the amount of energy contained in a single bug agent is set to 5.5 energy units, `bugs-per-patch` which represents the number of bugs which sprout from a newly grown patch of green grass is set to 2, and `grass-grow-rate` which represents the frequency with which a bare patch grows grass and with which a grassy patch grows dandelions is set to 0.033. Thus, at each time step, there is a 3.3% chance a black patch will be come green, and that a green patch will become yellow.

A SETUP procedure is used to initialize the rabbit, ferret, and skink populations. Within the SETUP procedure, first all variables of all global variables, turtles, and patches are cleared, all agents are terminated, all plots and output are cleared, and the time steps are reset to zero (Hint: see the `clear-all` command in the NetLogo Dictonary). Then each patch executes the GROW-GRASS submodel. Next, the initial population of rabbits is created with a population size of the global parameter `initial-num-rabbits`. All rabbits are given the same color and shape, but each is placed randomly within the model world, and given a random integer amount of `energy` in the range $[1, 10]$. Then, the initial population of ferrets is created with a population size of the global parameter `initial-num-ferrets`. All ferrets are given the same color and shape (different from the rabbits), but each is placed randomly within the model world, and given a random integer amount of `energy` in the range $[1, 10]$. Next, the initial population of skinks is created with a population size of the global parameter `inital-num-skinks`. All skinks are given the same color and shape (different from the rabbits and ferrets), but each is placed randomly within the model world, and given a random integer amount of `energy` in the range $[1, 10]$.

Input Data

There is no data being read into the model, instead it is built primarily by describing the biology of the system.

Submodels

The following submodels are included:

Grow-Grass This submodel is called by a single patch. A random number in the range $[0, 1]$ is generated, if it is below the value of `grass-grow-rate`, then the patch will grow grass if the patch is currently black and will grow dandelions if the patch is currently green. If the patch is currently green,

- the patch color is reset to yellow,

- the value of `grass-energy` is increased by 2, and
- 1 bug is sprouted on the patch (the bug should be shaped as a circle, colored red, and set to a size of 30% of the patch size).

If the patch is currently black,

- the patch color is reset to green,
- the value of `grass-energy` is set to a random integer in the range $[1, 10]$, and
- `bugs-per-patch` bugs are sprouted on the patch (the bug should be shaped as a circle, colored red, and set to a size of 30% of the patch size).

Note that if the patch is yellow, nothing happens. Also, note that green patches must be checked first. If black patches are checked first, then when we check the green patch next any newly green patches will become yellow in the same time step.

Grass-Death This submodel is called by a single patch. If the value of `grass-energy` of a single patch is greater than zero, then the value of `grass-energy` decreases by 0.5. Otherwise (i.e. if `grass-energy` is 0 or less), then

- If the patch is currently green, then the `countGtoB` counter is increased by 1.
- If the patch is currently yellow, then the `countYtoB` counter is increased by 1.
- The patch color is set to black (i.e., it is now bare land).
- All bugs on the patch die.
- The value of `grass-energy` for the patch is set to 0.

Note, this submodel provides a means for patches of grass to eventually disappear, even if there are only skinks, bugs, and ferrets left in the system, none of which consume grass.

Move This procedure is called by a single agent. The `heading` variable of the agent is reset to within 50° of its current heading, then the agent is moved forward one patch length. Lastly, the `energy` variable of the agent is decreased by 0.5 energy units.

Eat-Grass This submodel is called by a single rabbit agent, and what occurs depends on the color of the patch. If the patch is bare land (■), then nothing happens. If the patch contains grass (🟩), then the counter `countGtoB` is increased by 1, the color of the patch is set to black, the energy of the rabbit agent is increased by the amount of the `grass-energy` of the patch, and all the bugs currently residing on the patch die. If the patch contains grass with dandelions (🟨), then the counter `countYtoB` is increased by 1, the color of the patch is set to black, the energy of the rabbit agent is increased by the amount of the `grass-energy` of the patch plus 1 (the dandelions have extra energy), and all the bugs currently residing on the patch die.

Eat-Prey This submodel is called by a single ferret agent. The will eat if the is an appropriate prey source (rabbit or skink) within a 2-patch radius, otherwise the ferret does not eat in this time step. If there is an appropriate prey source, then one is selected to by the prey item. If the prey item is a skink on a patch with `grass-energy` > 0, then with a 20% probability the ferret consumes the skink (gains its energy), and the skink dies. if the prey item is a skink on a bare land patch or is a rabbit, then there is an 80% probability the ferret will consume the prey (gain its energy), and the prey dies.

EAT-BUG This procedure is called by a skink on a patch that contains 1 or more bugs. One of the bugs on the patch is selected to be the skink's meal (saved as a local variable called `meal`), then the energy of the skink increases by the amount of `bug-energy`, and the `meal` dies.

REPRODUCE This procedure is called by a single agent. If the agent is a skink and has `energy` > 3 or a ferrert and has `energy` > 8, then the `energy` of the agent is split in half, and an offspring (with all the same values for state variables as the parent) is hatched and moved forward 1 patch length (note, this will be in the same direction of the parent's `heading`). If the agent is a rabbit and has `energy` > 2, then the `energy` of the agent is split in half, and 2 offspring are hatched. The `heading` direction of each offspring is set to a random integer in the range $[0, 360]$, and the offspring moves forward 1 patch length.

DEATH This procedure is called by a single agent. If the energy of that agent is less than 0, then the agent dies.

The Assignment

1. Using the ODD description on the previous pages to construct an ABM using NetLogo. The NetLogo file should have code that is well commented, and an interface that uses input boxes for the input of initial population sizes and global parameters, appropriate buttons for running the SETUP and GO procedures, and the graphs described in the Observation section.

   In the GO button, use the code

   ```
   go
   if ticks > 300 [stop]
   ```

   to halt a simulation if it reaches 300 ticks.

2. Open up a Word document and save it using the naming convention `FinalProject_LastName.docx`. For the following questions, write responses and past any graphs into this document. This document should be well formatted, with a title and numbering to indicate which item is being answered. Any figures that you paste into the document should include a caption describing what is being shown, including initial population sizes of the rabbits, ferrets, and skinks. To give an image a caption, right click on the image and select "Insert Caption".

3. Run the model several times starting with 0 rabbits, 10 ferrets, and 150 skinks. Let the simulation continue until there are no skinks and ferrets left or until you have exceeded 300 time steps.
   (a) Take a screens shots of the resulting population graph for one or two of the simulations (include these in your project write up). You only need multiple screen shots if you get qualitatively different results (i.e.

both skinks and ferrets coexist in one simulation, but all the ferrets die in another simulation). Describe what sort of dynamics you observe (refer to the graphs as needed).

(b) Export the data from the Landscape Change graph and save it in as a `*.csv` file. Open the file in Excel, and calculate the average value of `rateGtoB` over all 300 ticks, and the average value of `rateYtoB` over all 300 ticks. Report these values in your write up. Repeat this process two more times. Thus, you should have three values for the mean `rateGtoB`, and three value for the mean `rateYtoB`. Use a table in your write up to report these six values.

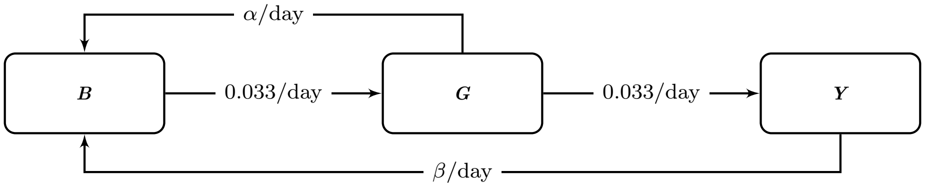

Figure 3: Flow diagram describing the landscape change in the ABM where B represents bare land (■), G represents grass (🟩), and Y represents grass with dandelions (🟨).

(c) Figure 3 show a flow diagram of the landscape change occurring in the ABM you have built. We know that on average 3.3% of black patches become green patches each time step (one day), and that on aveage 3.3% of green patches become yellow patches each time step. We can use the data that we generated from the ABM to determine the rate at which green patches become black patches (α = `rateGtoB`), and the rate at which yellow patches become black patches (β = `rateYtoB`). From part (b), you generated three different sets of (α, β) pairs. Use these values to create three different matrix models representing the flow diagram in Figure 3. For each model, determine the dominant eigenvalue of the system, if the system oscillates or reaches a structural equilibrium, and if it does reach a structural equilibrium determine what proportion of the model world is in each class at that structural equilibrium. Consider using a table to report your findings.

(d) Run 5 simulations and report the proportion of patches of each color at the end of each simulation (use a table to report these values). Use can use commands like

```
show (count patches with [pcolor = black]) / (count patches)
```

in the Command Center on the Interface tab in NetLogo to find these values. Given your results from part (c), would you classify your matrix models as good predictors of what will happen within your ABM?

4. Run the model several times starting with 200 rabbits, 10 ferrets, and 150 skinks. Let the simulation continue until there are no skinks and ferrets left or until you have exceeded 300 time steps.

   (a) Take a screens shots of the resulting population and landscape change graphs for some of the simulations (include these in your project write up). You only need multiple screen shots if you get qualitatively different results (i.e. both skinks and ferrets coexist in one simulation, but all the ferrets die in another simulation). Describe what sort of dynamics you observe (refer to the graphs as needed).

   (b) Describe why following steps 3(b) - 3(d), would likely not produce a reliable matrix model of the landscape change (refer to the graphs as needed).

References

[1] G. Norbury. Conserving dryland lizards by reducing predator-mediated apparent competition and direct competition with introduced rabbits. 2001. *Journal of Applied Ecology*, 38(6): 1350-1361.