# Chapter 5 Exercise Hints and Solutions

*Agent-based and Individual-Based Modeling: A Practical Introduction, 2nd Edition*

## Exercise 1

Versions of the Butterfly Model with the changes described in sections 5.2, 5.4, and 5.5 are available with the instructor materials.

The first step on page 71 asks why you should *not* put the new code to read the elevation input file within the `ask patches` statement. This is a very common mistake of beginners: thinking that anything related to patches must be within an `ask patches []` statement, and anything related to turtles must be within an `ask turtles []` statement. Instead, students need to understand that `ask patches` makes *each patch* execute the code inside the following brackets. If they put the code to read the input file and set elevations inside an `ask patches []` statement, then *each patch* will repeat the whole process of opening the file and setting *all* the patch elevations. You will recognize this problem because it takes an extremely long time for `setup` to finish.

### Exercise 2

The button should simply include, in its "Commands": `clear-drawing`

### Exercise 3

Butterflies do not cross the edge of the world because they move by jumping to a neighbor patch—either by using the primitive `uphill`, or by picking a patch from the agentset returned by the primitive `neighbors`. When world wrapping is turned off, the primitives `neighbors` and `uphill` do not include patches across the edge of the world.

You can prove this by right-clicking on a boundary patch to "inspect" it, then entering `show count neighbors` in the patch agent monitor's command window (explained in Section 6.3.4). When world wrapping is off, `show count neighbors` produces an answer of 5 for boundary patches and 3 for corner patches. If you then set world wrapping on, all patches have 8 neighbors. (As with many of the exercises, our real objective is to teach students how to investigate things.)

### Exercise 5

Some ways to measure how closely clumped the butterflies are at the end of a simulation are to use these statements. They can be added to the `go` procedure code that is executed when the model stops, or just typed into the Command Center.

- `show mean [count (other turtles) in-radius 2] of turtles`
- `show count patches with [any? turtles-here]`

### Exercise 7

It should be clear from the procedure we ask students to follow that the purpose of making butterflies keep moving even when on a hilltop is to keep them from getting stuck on local

hilltops. Instead, the butterflies eventually move on toward the highest peaks, which is clearly better for finding other butterflies.

## Exercise 10

Here are steps for transforming the data file `Exercise10_ElevationData.txt` so it can be used in the Butterfly Model. A spreadsheet with these calculations is provided.

1. Open the file in a spreadsheet; it can be imported as a tab-delimited text file.
2. The first column is labeled "Easting", which means these are X coordinates that increase in the east direction. The second column is labeled "Northing", which means these are Y coordinates that increase in the north direction. That means the data will display correctly in NetLogo if the World is set up with the origin (0,0) in the bottom left corner.
3. Translate the Easting coordinates into X coordinates that start at zero. This requires simply subtracting the minimum value of Easting from each number in the Easting column. You can determine from the spreadsheet that the lowest value of Easting in the file is 10,500. Create a new column that contains the Easting values minus 10,500. The lowest X value should now be 0.
4. Translate the Northing coordinates into Y coordinates that also start at zero, in the same way. The lowest Northing in the file is 47,500. You should now have a column of X values that range from 0 to 2010, and a column of Y values that range from 0 to 1530.
5. It should be clear now that these data had a spatial resolution of 10 units: when sorted in ascending order, the X and Y coordinates have values of 0, 10, 20, 30 etc. For NetLogo, we need the values to increase by 1, not by 10. Change the spatial resolution by creating new columns that contain the X and Y coordinates divided by 10. (This is equivalent to changing the units of the coordinates from meters to 10 meters: one patch is 10 meters across.)
6. Now, you should have a column of X values that range from 0 to 201, and a column of Y values that range from 0 to 153. Copy these new coordinate values to a clean spreadsheet page, and paste in the column of elevation values as the third column. Delete the header row because the NetLogo code cannot read it. Save this page as a tab-separated plain text file that will be your new input file for NetLogo.
7. The NetLogo program needs to be modified so that (a) the World is set up with `max-pxcor` = 201 and `max-pycor` = 153, and (b) the `setup` procedure includes the name of the new input file.