

*Lab Assignment 7*

**Loggerhead Sea Turtle Model**

*Written by*

Prof. Erin N. Bodine

*Prepared for*

Math 214: Discrete Math Modeling (Fall 2014)

Rhodes College

PreLab Assignment Due: **Thursday, 10/23/2014**

Lab Assignment Due: **Tuesday, 10/28/2014 by 11:59 pm**

Recall the American bison Leslie matrix model,

$$\mathbf{x}(t+1) = \begin{bmatrix} 0 & 0 & 0.42 \\ 0.60 & 0 & 0 \\ 0 & 0.75 & 0.95 \end{bmatrix} \mathbf{x}(t),$$

where  $x_1(t)$  is the number of female calves with age less than 1 year at time  $t$ ,  $x_2(t)$  is the number of female yearlings with age between 1 and 2 years at time  $t$ , and  $x_3(t)$  is the number of female adults (age greater than 2 years) at time  $t$ . In this model, the time step is one year.

We can use matrix algebra to determine index of imprimitivity and the dominant eigenvalue. If the index of imprimitivity is 1 then we can also find the normalized eigenvector representing the equilibrium structure (as we did in Lab 6). In the MATLAB Skills portion of this lab you will learn how to use a combination of built-in MATLAB function to find the index of imprimitivity, and how to use an *if-statement* in MATLAB.

## MATLAB Skills

### Calculating the Index of Imprimitivity

1. Open MATLAB and navigate the current directory to the location where you will save your work.
2. Download the L07Ex01.m file from Moodle, and open the file in MATLAB.

L07Ex01.m

```

1  % Lab 7 Example 1 by Prof Bodine
2  % Examine American Bison Leslie Matrix model
3  clear all
4
5  L = [0 0 0.42; 0.60 0 0; 0 0.75 0.95];          % Leslie matrix
6
7  % Determine if the population is oscillating
8  lambda = eigs(L); % all eigenvalues of L
9  a = abs(lambda); % take the absolute value of each eigenvalue
10 m = max(a);      % find the dominant eigenvalue
11 diff = abs(a - m); % calc diff btw each eigenvalue and the dom eigenvalue
12 tol = 1e-10;     % set the tolerance for what you accept as 0
13 d = length(find( diff < tol )); % calculate index of imprimitivity

```

3. Now, let us take a moment to understand everything within this script.

- (a) The first three lines give a description of the file in comments and use the `clear` command to erase the values of all variables and parameters from the workspace. If you do not remember how this command works, please review Lab 2, page 3.
- (b) Line 5 defines the matrix `L`.

- (c) Lines 7–13 are used to determine the index of imprimitivity of the Leslie matrix  $L$ . Note, if you wish to see what the output for any of these lines of code looks like, remove the semicolon from the end of the lines before running the code.
- (i) Line 8 saves all the eigenvalues of  $L$  in a vector called `lambda` using the `eigs` function that was introduced in Lab 6.
  - (ii) Recall the dominant eigenvalue is the one with the greatest absolute value (or magnitude). Thus, in line 9 we take the absolute value of each element in the array `lambda`, and save the result as `a`. Then, in line 10 we use the function `max` to obtain the largest absolute value of `lambda`.
  - (iii) Recall, the index of imprimitivity is the number of eigenvalues (including the dominant eigenvalue) which have the same absolute magnitude as the dominant eigenvalue. Let  $\lambda_1$  be the dominant eigenvalue, and  $\lambda_i$  be another eigenvalue such that

$$|\lambda_1| = |\lambda_i|, \quad \text{then} \quad |\lambda_1| - |\lambda_i| = 0.$$

Now, due to the limitations of machine precision (how precise a number is in the memory of a computer), sometimes when you subtract two numbers that are suppose to equal you end up with a very very small positive or negative number. For example, in the command window

```

Command Window
>> 0.5 - 0.4 - 0.1
ans =
    -2.7756e-17

```

If you subtract 0.4 and 0.1 from 0.5, you should end up with zero, but due to limitations of machine precision the calculated value is not zero. It is very close to zero ( $-2.7756 \times 10^{-17}$ ), but the computer has not calculated it to *be* zero. To get around this, we calculate the absolute value of the difference between each value in `a` and the value of `m` (see line 11), and then check to see if each of those values are less than some small number which we will consider equivalent to zero.

- (iv) In line 12 we set `tol` to be a small number which we will consider equivalent to zero. In the example, we have `tol` set to  $1 \times 10^{-10}$ .
- (v) In line 13 we calculate the index of imprimitivity. The function `find` allows you to search an array (vector or matrix) for values that meet certain criteria (in this case when `diff < tol`). The output of the `find` function are the indices (positions in the array) of the values which do meet the criteria. If you are curious what this output looks like, after you have run the code once, in the command window type

```

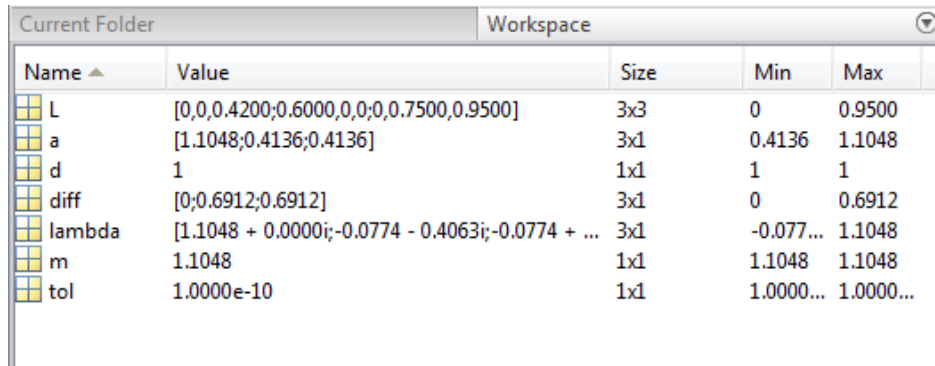
Command Window
>> find(diff < tol)
ans =
     1

```

This means that the only entry in `diff` that was less than `tol` was in entry 1 of `diff`. If there were more entries which met the criteria we would see a array of indices as the output. Note, that the

number of elements in the output of `find(diff < tol)` represents the number of eigenvalues of  $L$  which have the same absolute magnitude as the dominant eigenvalue. Thus, we take the length of the array output by `find(diff < tol)` to determine index of imprimitivity.

- (vi) Take another moment to review L05Ex01.m. **Do you understand what each line of code is doing?** If not, now is the time to stop and ask.
- (d) Run the script file. If you have not removed any of the semicolons, nothing will print out. However, you can see what values have been assigned to each variable the script created by examining the Workspace window.



Name	Value	Size	Min	Max
L	[0,0,0.4200;0.6000,0,0;0,0.7500,0.9500]	3x3	0	0.9500
a	[1.1048;0.4136;0.4136]	3x1	0.4136	1.1048
d	1	1x1	1	1
diff	[0;0.6912;0.6912]	3x1	0	0.6912
lambda	[1.1048 + 0.0000i;-0.0774 - 0.4063i;-0.0774 + ...	3x1	-0.077...	1.1048
m	1.1048	1x1	1.1048	1.1048
tol	1.0000e-10	1x1	1.0000...	1.0000...

Figure 1: View of Workspace window after running L07Ex01.m script.

## If-Statements

The idea of an if-statement is that you can check to see if some condition is true, if it is then you will execute one set of command, but if it is not then you will execute another set of commands.

1. From Moodle, download the script L07Ex02.m.

```

1  % Lab 7 Example 1 by Prof Bodine
2  % Examine American Bison Leslie Matrix model
3  clear all
4
5  L = [0 0 0.42; 0.60 0 0; 0 0.75 0.95];      % Leslie matrix
6
7  % Determine if the population is oscillating
8  lambda = eigs(L); % all eigenvalues of L
9  a = abs(lambda); % take the absolute value of each eigenvalue
10 m = max(a);      % find the dominant eigenvalue
11 diff = abs(a - m); % calc diff btw each eigenvalue and the dom eigenvalue
12 tol = 1e-10;     % set the tolerance for what you accept as 0
13 d = length(find( diff < tol )); % calculate index of imprimitivity
14
15 % Print out results

```

```
16 if d > 1
17     fprintf('Oscillates with a period of %d\n',d)
18     fprintf('Dominant eigenvalue is %4.3f\n',m)
19 else
20     [V,D] = eigs(L,1);
21     normV = V./sum(V);
22     fprintf('Dominant eigenvalue is %4.3f\n',D)
23     fprintf('Population structure at equilibrium is \n')
24     fprintf('  %4.3f\n',normV)
25 end
```

2. Now, let us take a moment to understand everything within this script.

- (a) First, note that lines 1–13 are copied from the script file `L07Ex01.m`.
- (b) Lines 16–25 contains an if-statement. The condition of the if-statement (`d > 1`) is given in line 16. If the index of imprimitivity is greater than 1, then the script will execute lines 17–18 and print out the period of the oscillations along with the dominant eigenvalue. If, on the other hand, the index of imprimitivity is not greater than 1 (which would mean it is 1), then lines 20–24 will execute. Note lines 20–21 calculate the normalized eigenvector corresponding to the dominant eigenvector, and lines 22–24 print out the dominant eigenvalue, and the population structure at equilibrium.

Note that the command `else` on line 19 indicates the end of the commands to be executed if the condition is true, and the beginning of the commands to be executed if the condition is false.

Lastly, like with a for-loop, we end an if-statement with the `end` command (see line 25).

3. Run the script file. You should see the following in the command window

```
Command Window
>> L07Ex02
Dominant eigenvalue is 1.105
Population structure is
    0.240
    0.130
    0.630
```

This means that the population does not oscillate and is growing (dominant eigenvalue  $> 1$ ), and that at equilibrium 24% of the American bison herd are calves, 13% are yearlings, and 63% are adults.

## PreLab Assignment

A herd of carefully managed beef cattle can be divided up into calves (less than one year old), yearlings (between 1 and 2 years old), and adults (older than 2). Only the adults birth calves. On average each adult female produces 1.2 female calves per year. The survival rate of the calves and the yearlings is 90%. However, after the two-year-old adults give birth, they are “processed” into beef.

- (a) Construct the Leslie matrix describing the herd of beef cattle.
- (b) Construct a script file similar to L07Ex02.m to determine if the population will approach a structural equilibrium or if it will cycle.

Name your script file `PLab07_YourLastName.m`.

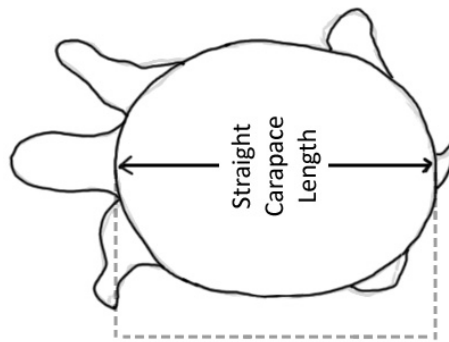
Make sure your script file is running without error and producing the output you expect before continuing to the next step.

- (c) In the command window type `diary PLab07_YourLastName.out` and press Enter. Run your script file, then typing in to the command window `diary off` and press Enter. Open up the diary file and make sure it has recorded the output of your script file.
- (d) Submit your script file and your diary file to Moodle. In the textbox on Moodle answer the following equations:
  - (a) Will the population approach a structural equilibrium? Why or why not?
  - (b) Is the population size stable? Should the manager of the herd worry about having a growing or declining population of cattle? Explain.

## Lab Assignment



(a) A loggerhead sea turtle.



(b) Diagram of straight carapace length measurement.

In 1987, Crouse, Crowder, and Caswell proposed a matrix model to simulate the population dynamics of loggerhead sea turtles.<sup>1</sup> The model is similar to a Leslie matrix model, but the individuals in each stage do not necessarily age into the next stage at each time step if they do not die. Instead, some proportion can stay in each stage at every time step. This is because the time step is one year, but each stage represents multiple years (see Table 1).

The stage-structured model proposed by Crouse, et al. used the seven age classes given in Table 1, however, since each class could represent more than one year, not only does the model have parameters for the fecundity of the  $i^{th}$  stage ( $F_i$ ), and the probability of surviving and growing to the next stage ( $G_i$ ), but it also includes the probability

<sup>1</sup>To read a copy of the original paper, visit <http://www.d.umn.edu/~thrabik/Crouse%20et%20al.%201987.pdf>.

**Table 1:** Stage-based life table for loggerhead sea turtles from [1].

Stage Number	Class Description	Straight Carapace Length (cm)	Approximate Age (yrs)	Annual Survivorship	Fecundity (# eggs/yr)
1	eggs, hatchlings	< 10	< 1	0.6747	0
2	small juveniles	10.1–58.0	1–7	0.7857	0
3	large juveniles	58.1–80.0	8–15	0.6758	0
4	subadults	80.1–87.0	16–21	0.7425	0
5	novice breeders	> 87.0	22	0.8091	127
6	1st-yr remigrants	> 87.0	23	0.8091	4
7	mature breeders	> 87.0	24–54	0.8089	80

of surviving and remaining in the same stage ( $P_i$ ). The Leslie-like matrix for the model is

$$\mathbf{A} = \begin{bmatrix} P_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 \\ G_1 & P_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & G_2 & P_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & G_3 & P_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_4 & P_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_5 & P_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & G_6 & P_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 127 & 4 & 80 \\ 0.6747 & 0.7371 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0486 & 0.6611 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0147 & 0.6907 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0518 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8091 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8091 & 0.8089 \end{bmatrix}.$$

Notice that for each stage  $G_i + P_i$  is equal to the annual survivorship of that class (shown in Table 1). The matrix model is then given by  $\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t)$ .

- Construct a script file in MATLAB, saved as `Lab07_YourLastName.m`, which accomplishes the following tasks in the order given:
  - Erase the values of all variables and parameters stored in the workspace.
  - Create the matrix  $\mathbf{A}$  (as given above).
  - Generate a plot of the number of female turtles in each age class over 100 years if you start with only 200 novice breeders (half of which are female). Note, to do this you will have to generate solutions first.
  - Determine if the population is oscillating by finding the index of imprimitivity. Use an if-statement to print out the results. If the index of imprimitivity is greater than 1 print out the period of the oscillations and the value of the dominant eigenvalue. If the index of imprimitivity is not greater than 1, then print out the value of the dominant eigenvalue and the population structure at the structural equilibrium.
- What happens to the population over the long term? How does the long term behavior of the population relate to the value of the dominant eigenvalue and the index of imprimitivity?
- Increase the value of  $G_2$  or  $G_3$  (or both) to obtain a different outcome. Remember, that the total survivorship of any particular class ( $G_i + P_i$ ) should not exceed 1. You should not have to greatly alter the values of  $G_2$  or  $G_3$  to obtain a qualitatively different long-term behavior outcome.
- Submit:**

- 1 diary file containing the output of running the script with the original model and with your modified model (`Lab07_YourLastName.out`)
- 1 script file (`Lab07_YourLastName.m`)
- 2 image files: one showing the graphical output of running the script with the original model and one generated from running the script with your modified model (`Lab07a_YourLastName.jpg` and `Lab07b_YourLastName.jpg`)
- In the text box on Moodle, answer question 2 and describe what sort of recommendations you would make to a group trying to conserve the loggerhead sea turtles based on your results from part 3.

## References

- [1] D.T. Crouse, L.B. Crowder, and H. Caswell. 1987. A Stage-Based Population Model for Loggerhead Sea Turtles and Implications for Conservation. *Ecology*, 68(5): 1412–1423.