Lab: Discrete Models: Stage Classes and Systems

1 A Few Basic Concepts

In the previous lessons we have learned about discrete models, how to simulate them and figure out the long-term behavior of these models through fixed points and stability analysis. Now let us consider a species which has multiple stage (or age) classes. As an example, consider an insect that has two distinct stage classes: larva and adult. Let us also assume that only the adults can produce offspring while the larval stage is incapable of reproducing. We could model these insects as before, except how will we separate out those of the population who can reproduce and those who are too young? A common modeling technique is to split the population into two stages (or age) classes and model each of the stages (or ages) separately. This type of modeling is known as a stage (or age) class model or a Leslie model or Leslie matrix, named after P. H. Leslie who invented/discovered this model in 1945. We will start by looking at an example of a population with a "young" stage class and an "adult" stage class. As you will see, we can extend this idea and modeling technique to as many stages as necessary to model the population.

2 Stage-Class Models

This is very similar to what we have previously worked with before. Granted there are some differences which we will have to deal with, but let us start at the beginning. For our example population we will have two stages classes, young and adult. For this model we will assume the young stage lasts for a year and the adult stage lasts for several years and then eventually the organism dies. Now we have two equations. This can be mathematically written as:

$$\begin{aligned} Y_{t+1} &= bA_t \\ A_{t+1} &= mY_t + rA_t \end{aligned}$$

where Y_t is the young population at time t and A_t is the adult population at time t. The parameter b is the birth rate of the young from the adult population, the parameter m is the rate at which the young mature into adults and parameter r is the surviving adult population from the previous year. This particular model can also be written in matrix form:

$$\begin{bmatrix} Y \\ A \end{bmatrix}_{t+1} = \begin{bmatrix} 0 & b \\ m & r \end{bmatrix} * \begin{bmatrix} Y \\ A \end{bmatrix}_t$$

As you may have guessed this form is why the Leslie Model is also known as the Leslie Matrix. Let us begin with a simulation. The code is very similar to the programs we have worked with before. Be sure to save this MATLAB file as"leslie1.m".

```
% Your Name
% Today's Day
% Course Number
% Leslie Model 1: From lecture notes

function [] = leslie1()

% Number of time steps (years)
n = 100;

% Parameter and Initial Conditions
b = 2; % birth rate
m = 0.3; % maturation rate from young to adult
r = 0.5; % return rate (survival rate) for the adult population
initY = 10; % Initial Young population
initA = 50; % Initial Adult population
initYear = 0; % Initial Year

%Solution Populations and Time
Y = zeros(1,n+1); % The n+1 is to store our initial populations plus the ones we find
A = zeros(1,n+1);
Year = zeros(1,n+1);

%Set inital conditions
Y(1) = initY;
A(1) = initA;
Year(1) = initYear;

% Equations
for i=1:n
    Y(i+1) = b*A(i); % Young Population Equation
    A(i+1) = m*Y(i) + r*A(i); % Adult Population Equation
    Year(i+1) = i; % Keeping track of the time (years)
end

% Plot
figure(1)
clf
hold on
plot(Year, Y, 'r-', 'linewidth', 3)
plot(Year, A, 'b.-', 'linewidth', 3)
xlabel('Time (years)')
ylabel('Population')
title('Leslie Model with two stage classes')
```

set(gca, 'fontsize', 20)
hold off

2.1 Fixed Point(s) of Discrete Systems

Now that we have a simulation you may notice that this model only has one fixed point, the trivial state. Note that the use of the the word "point" refers to the long-term behavior of both populations, young and adult. In this case, perhaps "state" would be a better word to use. How can we find this mathematically? We follow the same procedure as before, though now we need to solve a system of equations. Remember to remove the subscripts on the variables and set both equations to zero before solving the system.

```
>> clear Y A b r m Yfixed Afixed
>> syms Y A b r m Yfixed Afixed
>> [Yfixed,Afixed]=solve('Y-b*A', 'A-m*Y-r*A', 'Y,A')
```

This gives us only one fixed state, the trivial state. Now that we have a fixed state (or steady state) we can look at the stability. This will give us a few more problems than before.

3 Stability of Fixed Points

The basic idea of stability is the same as before. We want to know how small perturbations will affect each fixed "point" or "state". We will use MATLAB heavily to find the stability, but before we can do that we need to introduce two more concepts in mathematics.

3.1 The Jacobian

The first concept is called a Jacobian, which in short is a matrix of derivatives. Now that we have multiple functions of more than one variable, we need to figure out which derivatives we are taking of what and with respect to what. To explain this, let's refer back to our example young and adult population:

$$\begin{aligned} Y_{t+1} &= bA_t \\ A_{t+1} &= mY_t + rA_t \end{aligned}$$

Ignoring the subscripts, if we take the derivative of the first function bA with respect to Y we get 0, while the derivative of bA with respect to A equals b. Now if we take the derivative of the second function $mY + rA$ with respect to Y we get m and finally the derivative of $mY + rA$ with respect to A equals r. The Jacobian gives us a better way of organizing these derivatives:

$$\begin{bmatrix} 0 & b \\ m & r \end{bmatrix}$$

Of course, MATLAB has a built in function to find the Jacobian! To use this function labeled **jacobian** you need to enter in the equations from the model in order, separated by semicolons and without subscripts. Here is an example for our two-stage class model:

```
>> clear A Y b m r
>> syms A Y b m r
>> jacobian([b*A; m*Y+r*A],[Y A])
```

This is the exact answer we just found on our own. Since we know the parameter values, we can substitute them in before we find the jacobian as well!

```
>> clear A Y
>> syms A Y
>> jacobian([2*A; 0.3*Y+0.5*A],[Y A])
```

3.2 Eigenvalues

The second important concept to be introduced are eigenvalues. For our purposes, we will use these values to determine if the fixed "point" or fixed "state" is stable or not, but we will not go over the mathematics behind eignevalues. Instead we will use MATLAB to find these values. We want to find the eigenvalues of the jacobian, so we will need to find and label the jacobian matrix first and then find the eigenvalues of that matrix using the built in MATLAB function **eig**.

```
>> clear A Y b m r
>> syms A Y b m r
>> jac = jacobian([b*A; m*Y + r*A],[Y A])
>> eig(jac)
```

This returns your eigenvalues, though they are buried within a vector. Each entry in the vector is one of the eigenvalues for the system. In this case we have two eigenvalues. Once MATLAB has found the eigenvalues for the system, you need to take the **absolute value** of each eigenvalue. Similar to before, the fixed state is:

- **Stable** if the **largest** absolute value is LESS THAN one
- **Unstable** if the **largest** absolute value is GREATER THAN one
- **Undetermined** if the **largest** absolute value EQUALS one

For our system with the young and adult populations, the only fixed point (the trivial state) depends on the values of the parameters b, m, and r. The eigenvalues for this system are

$$\frac{r + \sqrt{r^2 + 4mb}}{2}$$

and

$$\frac{r - \sqrt{r^2 + 4mb}}{2}$$

and finding the absolute values of these expression may not be fun. Once again, if you know the parameter values, MATLAB will be able to return numerical values for the eigenvalues.

```
>> clear A Y
>> syms A Y
>> jac2 = jacobian([2*A; 0.3*Y+0.5*A],[Y A])
>> eig(jac2)
```

When we put in the parameter values $b = 2$, $m = 0.3$ and $r = 0.5$, MATLAB gives us eigenvalues of 1.0639 and -0.5639. In absolute values we have 1.0639 and 0.5639. Since 1.0639 is larger than one, this state is unstable. Biologically this is good for our system, as this population will not tend toward extinction!

4 General Notes on Discrete Systems

You now know enough with programming and MATLAB to work with large discrete systems. Each time you add another "age/stage" class be sure to alter your program, the Jacobian and find new eigenvalues. Let us put all this knowledge to use!

5 Practice Problems

1. Find the biologically relevant fixed points of the following system:

$$
\begin{aligned}
X_{t+1} &= aX_t + bY_t \\
Y_{t+1} &= cX_t + dY_t
\end{aligned}
$$

   with the following parameter values:

   (a) $a = 0.1$, $b = 0.2$, $c = 0.3$ and $d = 0.4$

   (b) $a = 0.1$, $b = 0.4$, $c = 0.2$ and $d = 0.3$

   (c) $a = 0.9$, $b = 0.5$, $c = 0.5$ and $d = 0.9$

2. Using the model and fixed points from the previous problem, find the stability of each fixed point.

3. Our two-stage class model (young and adult) had the form:

$$
\begin{aligned}
Y_{t+1} &= 2A_t \\
A_{t+1} &= 0.3Y_t + 0.5A_t
\end{aligned}
$$

   Now we have altered this model to include a third stage class (larvae, juvenile and adult):

$$
\begin{aligned}
L_{t+1} &= 2A_t + J_t \\
J_{t+1} &= 0.5L_t \\
A_{t+1} &= 0.3J_t + 0.5A_t
\end{aligned}
$$

   Answer the following questions:

   (a) Biologically describe this system.

   (b) Find the fixed points and their stability.

   (c) How would you expect the simulation of the two-stage class model to differ from this three-stage class model?

   (d) How would you expect the simulation of the two-stage class model to be the same as the this three-stage class model?

   (e) Run a simulation of the two-stage class model with the following initial conditions: $Y_0 = 50$ and $A_0 = 0$.

   (f) Compare this to a simulation of the three-stage class with initial conditions: $L_0 = 50$, $J_0 = 0$ and $A_0 = 0$.

   (g) Were your expectations met?

4. Using the following model

$$X_{t+1} = \frac{X_t Y_t}{1 + X_t}$$
$$Y_{t+1} = 2X_t$$

answer the following questions:

(a) Find all fixed points.

(b) Which of these points (if any) are biologically relevant.

(c) Find the stability of each of the biologically relevant fixed points.

(d) Simulate this model starting with the following different initial conditions:

i. initial X value $= 0$ and initial Y value $= 0$

ii. initial X value $= 0.5$ and initial Y value $= 0.75$

iii. initial X value $= 2$ and initial Y value $= 3$

(e) Do your simulation results match your stability analysis? Why or why not?