

Lab Assignment 5

Stochastic Aquatic Succession

Written by

Prof. Erin N. Bodine

Prepared for

Math 214: Discrete Math Modeling

Rhodes College

Last Edited: April 16, 2018

Recall from our reference readings the example of ecological succession of coastal wetlands. In this example, we consider 100 hectares of wetlands, which at the initial time are always submerged. Every decade, 5% of the always submerged wetlands become seasonally submerged, and 12% of the seasonally submerged wetlands becomes always dry. All hectares that are dry remain dry.

In this lab we will learn how to use MATLAB to simulate a matrix model representing this example of ecological succession, as well as a few other examples. We will also examine what happens when rate of movement from one state to another is stochastic.

In this lab, we will be using MATLAB to help achieve these goals. Thus, as a part of this lab you will learn how to do the following in MATLAB:

- Enter vectors in MATLAB in either row form or column form.
- Enter matrices in MATLAB.
- Simulate a matrix model.
- Print output tables in the Command Window.
- Plot the dynamics of a matrix model over time.

MATLAB Skills

At this points, you already know how to do quite a bit in MATLAB, however there are a couple of new MATLAB skills you will need in order to complete this lab. As a part of the Prelab Assignment, read through the following sections. **Note, all examples files are posted on Moodle.**

Vectors in MATLAB

Let

$$\mathbf{x} = [2 \ 5 \ 2 \ 4 \ 6] \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} 4 \\ 7 \\ 5 \\ 8 \\ 11 \end{bmatrix}.$$

Since \mathbf{x} is represented as one row with many columns, we call \mathbf{x} a **row vector**, and since \mathbf{y} is represented as one column with many rows, we call \mathbf{y} a **column vector**. Note, that both vectors have the same number of elements. To enter these two vectors in to MATLAB we would use the commands

$$\mathbf{x} = [2 \ 5 \ 2 \ 4 \ 6] \quad \text{and} \quad \mathbf{y} = [4; 7; 5; 8; 11]$$

The semicolon indicates that the next entry should be placed in the next row of the vector. Try entering these two commands into the command window of MATLAB . Do your vectors appear as they should?

If you want \mathbf{x} to appear as a column vector, enter \mathbf{x}' into the command window. This creates what is called the **transpose** of the vector \mathbf{x} , which has the effect of transforming a row vector into a column vector. If you enter \mathbf{y}' into the command window, it will transpose the column vector \mathbf{y} into a row vector.

Matrices in MATLAB

The name MATLAB is actually short for Matrix Laboratory. One of Matlab's main features is that it quickly and easily handles and manipulates matrices.

1. The ecological succession of coastal wetlands described at the beginning of this lab can be modeled with the matrix equation,

$$\mathbf{x}(t+1) = \begin{bmatrix} 0.95 & 0 & 0 \\ 0.05 & 0.88 & 0 \\ 0 & 0.12 & 1 \end{bmatrix} \mathbf{x}(t), \text{ where } \mathbf{x}(t) = \begin{bmatrix} a(t) \\ s(t) \\ d(t) \end{bmatrix}$$

and $\mathbf{x}(t)$ represents the structure of the coastal wetlands at time step t , where $a(t)$ is the proportion of wetlands that is always submerged at time step t , $s(t)$ is the proportion that is seasonally submerged at time step t , and $d(t)$ is the proportion that is always dry at time step t . Let

$$\mathbf{T} = \begin{bmatrix} 0.95 & 0 & 0 \\ 0.05 & 0.88 & 0 \\ 0 & 0.12 & 1 \end{bmatrix}.$$

To enter \mathbf{T} into MATLAB we would use the command

$$\mathbf{T} = [0.95 \ 0 \ 0; \ 0.05 \ 0.88 \ 0; \ 0 \ 0.12 \ 1]$$

Note that the semicolon is again used to denote when to start the next row of entries. Enter the command into the command window and press Enter. Notice that the output displays as a matrix.

2. Suppose we want to retrieve the element of \mathbf{T} in the third row, second column. To do this we use the command `T(3,2)`. Enter this command into the command window and press Enter. In general, if we want to retrieve the element in the i^{th} row and j^{th} column we use the command `T(i,j)`. Try retrieving the element in the second row, first column.
3. What if you want to retrieve an entire column or row? If you wish to retrieve the i^{th} row, use the command `T(i,:)`, where the colon indicates *all columns*. If you wish to retrieve the j^{th} column, use the command `T(:,j)`. Here, the colon indicates *all rows*.

Simulate a Matrix Model

1. From Moodle, download the script file `L05Ex01.m`.

```

1  % Lab 5 Example 1 by Prof Bodine
2  % Simulates Coastal Wetlands Ecological Succession Model
3  clear all
4
5  % Transfer matrix
6  T = [0.95 0 0; 0.05 0.88 0; 0 0.12 1];

```

```

7  % Initial condition: a(0) = 1, s(0) = 0, d(0) = 0
8  x0 = [1; 0; 0];
9
10 % Generate solutions over 100 times steps
11 x(:,1) = x0;
12 for n = 1:100
13     x(:,n+1) = T * x(:,n);
14 end

```

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

- 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.
- Line 6 defines the matrix T , and line 8 defines the initial state of the coastal wetlands (all of the 100 hectares are always submerged and thus in the first state, a).
- In this code we generate solutions over 100 times steps. The solution at each time step gives the proportion of the coastal wetlands in each state (a , s , and d) at that time step, and is represented as a column vector

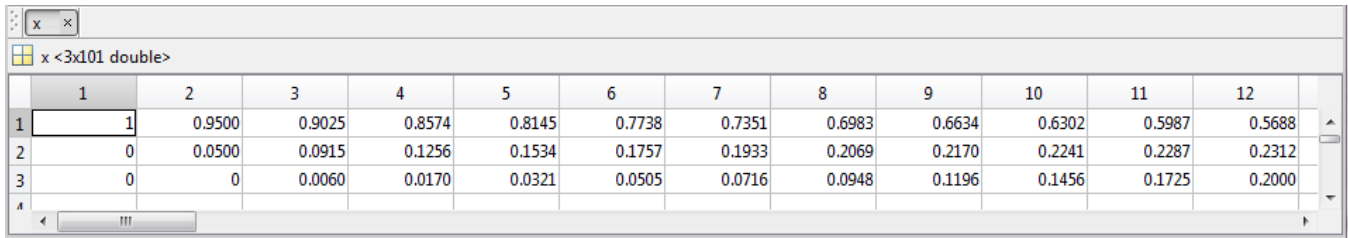
$$\mathbf{x}(t) = \begin{bmatrix} a(t) \\ s(t) \\ d(t) \end{bmatrix}.$$

In MATLAB, we save each of these column vectors as one column in a matrix called \mathbf{x} . In line 11, we save the initial condition as the first column of \mathbf{x} . Then in lines 12–14, we use a for-loop to generate the solutions at time steps 1 through 100. Each pass through the for-loop generates the solution at time step n , and saves that solution in the $n+1$ column of the matrix \mathbf{x} . Notice, $\mathbf{x}(:,n)$ indicates that all rows in column n of matrix \mathbf{x} are being retrieved.

- 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.
3. Run the script file. To see what the solutions are, double click on the \mathbf{x} variable listed in the Workspace window of MATLAB. You should see a window open that look like Figure 1. You can scroll to the right to see view more. Remember, each column of \mathbf{x} represents the “structure” of the coastal wetlands where the first row gives the proportion which is always submerged, the second row gives the proportion which is seasonally submerged, and the third row gives the proportion which is always dry.

Creating Output Tables using `fprintf` & a For-Loop

Viewing the elements of the matrix \mathbf{x} from the Workspace window can sometimes be a bit overwhelming if the matrix you are viewing is quite large. Sometimes, it is nice to view just a sampling of the matrix. We can print out the



The screenshot shows a MATLAB workspace window with a variable 'x' of type 'double' and size '3x101'. The first 12 columns of the matrix are displayed in a table format:

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0.9500	0.9025	0.8574	0.8145	0.7738	0.7351	0.6983	0.6634	0.6302	0.5987	0.5688
2	0	0.0500	0.0915	0.1256	0.1534	0.1757	0.1933	0.2069	0.2170	0.2241	0.2287	0.2312
3	0	0	0.0060	0.0170	0.0321	0.0505	0.0716	0.0948	0.1196	0.1456	0.1725	0.2000

Figure 1: Example of window that appears when you double click on the x variables listed in the Workspace window of MATLAB.

solutions at only the time steps we are interested in by using an `fprintf` statement inside a for-loop.

1. From Moodle, download the script file L05Ex02.m.

```

L05Ex02.m
1  % Lab 5 Example 2 by Prof Bodine
2  % Simulates Coastal Wetlands Ecological Succession Model
3  clear all
4
5  % Transfer matrix
6  T = [0.95 0 0; 0.05 0.88 0; 0 0.12 1];
7  % Initial condition: a(0) = 1, s(0) = 0, d(0) = 0
8  x0 = [1; 0; 0];
9
10 % Generate solutions over 100 times steps
11 x(:,1) = x0;
12 for n = 1:100
13     x(:,n+1) = T * x(:,n);
14 end
15
16 % Print table of solution for a selection of time steps
17 fprintf(' t      a      s      d\n')
18 fprintf('-----\n')
19 for n = [0 1 2 3 4 5 10 20 30 40 50 75 100]
20     fprintf('%3d %4.3f %4.3f %4.3f\n',n,x(1,n+1),x(2,n+1),x(3,n+1))
21 end

```

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

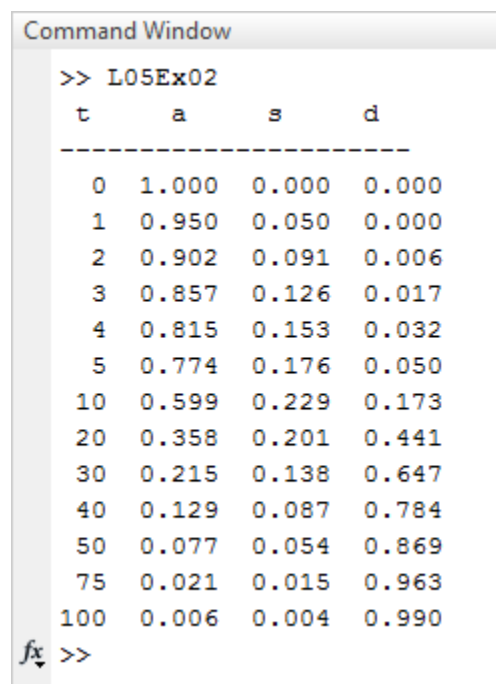
- (a) First, note that lines 1–14 are copied from the script file L05Ex01.m.
- (b) In lines 16–21 we print out a table of the solution for selected time steps. Lines 17 and 18 print out a header for the table, and lines 19–21 use a for-loop to fill in the table. Each pass through the for-loop fills in one line of the table, which represents the solution for one time step. Notice, that we are not printing

the solutions at all time steps, but only for a selection of time steps given in the list

```
n = [0 1 2 3 4 5 10 20 30 40 50 75 100]
```

in the set up of the for-loop.

- (c) In the `fprintf` statement on line 20, the `%3d` command is used to hold 3 spaces for the integer value of `n`, and `%4.3f` is used to hold 5 spaces (one digit before the decimal, a decimal, and three digits after the decimal) for the floating point values (real number values) of `x(1,n+1)`, `x(2,n+1)`, and `x(3,n+1)`. Notice, these retrieve the three values in the `n+1` column of `x` which represents time step `n`.
- (d) 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.
3. Run the script file. You should see some output printed in the command window (see Figure 3 for reference). What is happening to the structure of the coastal wetlands over time?



```
Command Window
>> L05Ex02
t      a      s      d
-----
0  1.000  0.000  0.000
1  0.950  0.050  0.000
2  0.902  0.091  0.006
3  0.857  0.126  0.017
4  0.815  0.153  0.032
5  0.774  0.176  0.050
10 0.599  0.229  0.173
20 0.358  0.201  0.441
30 0.215  0.138  0.647
40 0.129  0.087  0.784
50 0.077  0.054  0.869
75 0.021  0.015  0.963
100 0.006  0.004  0.990
fx >>
```

Figure 2: Output printed in the Command Window when the script file `L05Ex02.m` is run.

Generating Plots of Matrix Model Dynamics over Time

A picture says a 1000 words. Well, a graph of our solutions might tell us more about

1. From Moodle, download the script file `L05Ex03.m`.

```
----- L05Ex03.m -----
1  % Lab 5 Example 3 by Prof Bodine
2  % Simulates Coastal Wetlands Ecological Succession Model
```

```
3 clear all
4
5 % Transfer matrix
6 T = [0.95 0 0; 0.05 0.88 0; 0 0.12 1];
7 % Initial condition: a(0) = 1, s(0) = 0, d(0) = 0
8 x0 = [1; 0; 0];
9
10 % Generate solutions over 100 times steps
11 x(:,1) = x0;
12 for n = 1:100
13     x(:,n+1) = T * x(:,n);
14 end
15
16 % Plot solutions over time
17 plot(0:100,x)
18 xlabel('Time (decades)'); ylabel('Proportion of 100 hectares');
19 legend('a','s','d','Location','East')
```

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

(a) First, note that lines 1–14 are copied from the script file L05Ex01.m.

(b) Lines 16–19 create a graph with three curves: one curve is the proportion of wetlands always submerged over time, the second is the proportion that is seasonally submerged over time, and the third is the proportion that is dry over time. Notice, that the plot command in the script does not explicitly list the three curves, nor does it define the color for each curve.

(c) Line 18 produces axes labels for the graph.

(d) Line 19 produces a legend for the graph and places it in the middle, right portion of the graph. Other possible locations are given in Table 1.

(e) 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.

3. Run the script file. You should see some output printed in the command window (see Figure 3 for reference). What is happening to the structure of the coastal wetlands over time? You should observe, that by time step 100 (100 decades = 10,000 years), almost all of the wetlands are completely dry land.

Calculating Eigenvalues & Eigenvectors

1. From Moodle, download the script file L05Ex03.m.

Table 1: Possible locations to place a legend in a MATLAB plot.

Specifier	Location in Axes	Specifier	Location in Axes
North	Inside plot box near top	South	Inside bottom
East	Inside right	West	Inside left
NorthEast	Inside top right (default for 2-D plots)	NorthWest	Inside top left
SouthEast	Inside bottom right	SouthWest	Inside bottom left
NorthOutside	Outside plot box near top	SouthOutside	Outside bottom
EastOutside	Outside right	WestOutside	Outside left
NorthEastOutside	Outside top right (default for 3-D plots)	NorthWestOutside	Outside top left
SouthEastOutside	Outside bottom right	SouthWestOutside	Outside bottom left
Best	Least conflict with data in plot	BestOutside	Least unused space outside plot

```

1  % Lab 5 Example 4
2
3  % Transition matrix for aquatic succession
4  T = [0.95 0 0; 0.05 0.88 0; 0 0.12 1];
5
6  % Find the eigenvalues and eigenvectors
7  [V,D] = eig(T);
8  L = diag(D); % all the eigenvalues
9
10 % Find the dominant eigenvalue and corresponding eigenvector
11 imax = find(L==max(L));
12 Lmax = L(imax)
13 Vmax = V(:,imax);
14 normVmax = Vmax/sum(Vmax)

```

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

- (a) First, on line 4, we define the matrix for which we want to find the eigenvalues and eigenvectors.
- (b) Next, the `eig` command on line 7, calculates the eigenvalues (saved along the diagonal of the matrix `D`) and corresponding eigenvectors (stored as columns in the matrix `V`). The first column of matrix `V` is an eigenvector corresponding to the eigenvalue in position (1,1) in matrix `D`; the second column of matrix `V` is an eigenvector corresponding to the eigenvalue in position (2,2) in matrix `D`; etc. Note, that the corresponding eigenvectors are not necessarily normalized. That is, their elements may not sum to 1.

Note, if you change line 7 to `x=eig(T)`, the function `eig` only returns a list of the eigenvalues of `T`.

- (c) In line 8 we use the `diag` command in Matlab to extract the elements along the diagonal of the matrix `D`. Thus, `L` is a vector of our eigenvalues.

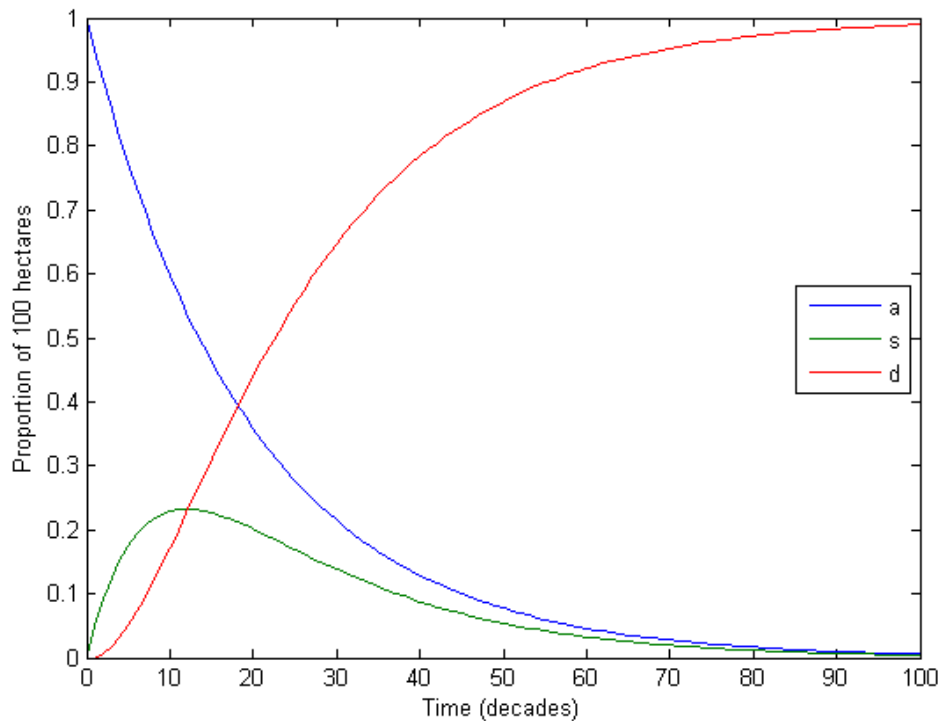


Figure 3: Plot produced from L05Ex03.m script.

- (d) Lines 10–14 make use the `find` command to determine the dominant eigenvalue and its associated normalized eigenvector. The `find` function in Matlab will return of indices of a one-dimensional array given the conditional statement in the parentheses. In this case, we want the indices of the element of the array `L` which is equal to the maximum element of `L` (i.e., the dominant eigenvalue). We save the index of the dominant eigenvalue as `imax`.

In line 12, we save the dominant eigenvalue as `Lmax`; notice we are accessing the `imax` element of the array `L`.

In line 13, we save the eigenvector associated with the dominant eigenvalue as `Vmax`; notice we are accessing the `imax` column of the 2-dimensional array `V`. Since the elements of this eigenvector may not sum to 1, we normalize the eigenvector in line 14. We do this by taking the vector `Vmax` and dividing by the sum of its elements `sum(Vmax)`.

Lab Group Discussion

Week 1:

1. When entering a matrix into Matlab, how do we indicate one row from the next?
2. Is the initial condition for a matrix model a real number, a row vector, or a column vector?
3. In L05Ex01.m, why is the variable `x` on lines 11 and 13 a 2-dimensional array instead of a 1-dimensional array?

4. In `L05Ex01.m`, what function does the colon have on lines 11 and 13?
5. In `L05Ex01.m`, what function does the colon have on line 12?
6. In `L05Ex02.m`, why do we include lines 17–18?
7. In `L05Ex02.m`, why do we need to place the `fprintf` statement inside a for-loop?
8. In `L05Ex02.m`, how could I change line 19 so that only results for every 5 decades printed out to the Command Window?
9. In `L05Ex02.m` on line 19, why do we use the column index `n+1` inside the array `x`?
10. In `L05Ex03.m`, change line 17 to `plot(x)`. How does this change the graph? Specifically, what is being graphed? What do the horizontal and vertical axes represent?
11. What is the transition matrix for the flow diagram given in the Week 1 Checkpoint Exercise?
12. For a matrix model describing the flow diagram given in the Week 1 Checkpoint Exercise, what does one step represent?

Week 2:

1. Will the `L05Ex04.m` script still work if the dimension of the array `T` are changed? For example if we had a model with a 4×4 or 7×7 matrix?
2. What are the diagonal elements of the matrix `T`? After running `L05Ex04.m`, type `diag(T)` in the command window and confirm that the `diag` function is taking the elements along the diagonal.
3. In `L05Ex04.m`, what does the matrix `V` represent?
4. In `L05Ex04.m`, what does the matrix `D` represent?
5. Run `L05Ex03.m`. In the command window, write a line of code to find the decade (i.e. the time step) at which the state `s` is maximized.
6. After running `L05Ex04.m` type `find(L==max(L))` in the command window. You should get an error message. What is wrong with this code?

Week 3: As you start the lab exercise this week you will need to recall how to generate random numbers drawn from a uniform distribution.

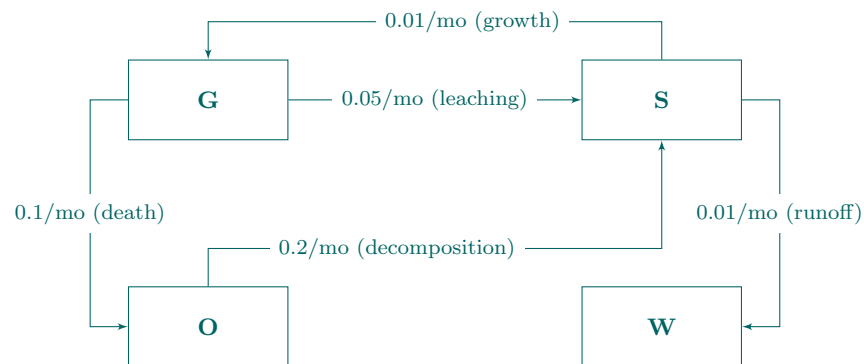
1. Write a line of code that generates one sample from the uniform distribution $\mathcal{U}[0.05, 0.25]$.
2. Write a for-loop which iterates 10 times, and in each iteration generates a random number drawn from $\mathcal{U}[0.05, 0.25]$ and stores that value in an array called `r`.
3. Matlab also provides the `randn` function as a way to sample from the normal distribution. Specifically, it samples from the normal distribution with mean 0 and variance 1, denoted $\mathcal{N}(0, 1)$. Note, variance is the standard deviation squared. To generate a random sample from $\mathcal{N}(a, b)$, we use `b*randn+a`, where `a` is the mean and `b` is the variance. Write a for-loop which iterates 10 times, and in each iteration generates a random number drawn from $\mathcal{N}(0.15, 0.1)$ and stores that value in an array called `r`.

4. Repeat #2, but generate 1000 samples and plot using a histogram. Does the distribution of the sampled values appear uniformly distributed?
5. Repeat #3, but generate 1000 samples and plot using a histogram. Does the distribution of the sampled values appear normally distributed?
6. Alter your code in #5 to use more or fewer bars to construct the histogram. Does the distribution of sampled values still appear normally distributed? What is a good number of bars to use in the histogram to best display the “bell shaped curve” of the data.

Checkpoint Exercise

Week 1: Strontium-90 is a radioactive by-product produced by the nuclear fission of uranium and plutonium. The atmospheric nuclear weapons tests of the 1950s and 1960s produces a substantial amount of strontium-90 as radioactive fallout and was dispersed into ecosystems around the world. When an organism ingests strontium-90 from contaminated food or water, most of the compound is excreted, but 20–30% becomes deposited in the organisms bones and bone marrow. This can lead to bone cancer, and leukemia. However, strontium-90 is also used a radioactive source for the radiotherapy used to treat some cancers and eye diseases, and as a heat source for thermoelectric devices used in navigational beacons, remote weather stations, and space vehicles.

Strontium-90 cycles through ecosystems, and this dynamic process can be describe by a matrix model as represented by the flow diagram below where the ecosystem is divided into grasses (G), soil (S), streams and waterways (W), and dead organic matter (O).



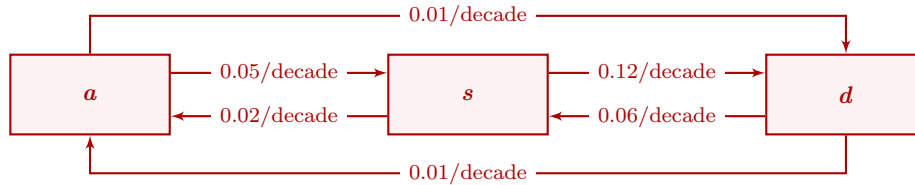
1. Determine the matrix model which represent the flow diagram above.
2. Write a MATLAB script which generates solutions for the model described by the above flow diagram over 5 years (i.e., 60 months) assuming 100% of the strontium-90 is in dead organic matter at month $t = 0$.

The print out the of proportion of strontium-90 in each state at the end of each year (i.e. every 12 months). The print out should have the appearance of a table with a header. Each row should correspond to the results at the end of a year, and columns should correspond to the states of the model.

Your script file should also produce a graph of each state over time. The graph should have clear axes labels and and a legend placed in a location where it does not overlap with the plotted curves.

3. **CHECKPOINT:** Verify that your code is running correctly by executing your script file for the lab tutor.

Week 2: Recall that in Complex Wetlands Succession Example where the movement along the flow diagram was not all in one direction. This dynamic process can be described by a matrix model as represented by the flow diagram below where the wetlands are divided into always submerged (a), seasonally submerged (s), and always dry (d).



1. Determine the matrix model which represent the flow diagram above.
2. Write a MATLAB script which generates solutions for the model described by the above flow diagram over 100 decades assuming 100% of land begins as always submerged.

The print out the of proportion of the wetlands in each state at the every 10 decades. The print out should have the appearance of a table with a header. Each row should correspond to the results at the end of a year, and columns should correspond to the states of the model.

Your script file should also produce a graph of each state over time. The graph should have clear axes labels and and a legend placed in a location where it does not overlap with the plotted curves.

You may recycle your code from last week's checkpoint for this exercise.

3. Modify your code to additionally find the dominant eigenvalue and its associated eigenvector.
4. **CHECKPOINT:** Verify that your code is running correctly by executing your script file for the lab tutor.

Lab Assignment

1. Modify the MATLAB script you created for the Week 2 Checkpoint Exercise so that the rate at which always submerged wetlands (a) become seasonally submerged wetlands (s) is sampled from $\mathcal{U}[0.01, 0.08]$. The rate should be re-sampled at each time step, which means the matrix must be redefined at each time step (i.e. within the for-loop). Remember, that each column in the matrix T must sum to 1.

Name your script file `Lab05_YourLastName.m`.

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

2. In the command window type `diary Lab05_YourLastName.A.txt` and press Enter. Run you script file 3 times. During one of the runs save the resulting figure using the names `Lab05_YourLastName.A.jpg`. Next, type into 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.
3. Further modify your MATLAB script so that the rate at which always dry wetlands (d) become seasonally submerged wetland (s) is sampled from $\mathcal{N}(0.6, 0.05)$. The rate should be re-sampled at each time step. Remember, that each column in the matrix T should sum to 1.

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

4. In the command window type `diary Lab05_YourLastName.B.txt` and press Enter. Run you script file 3 times. During one of the runs save the resulting figure using the names `Lab05_YourLastName.B.jpg`. Next, type into 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.

5. **Submit:**

- 1 script file (`Lab06_YourLastName.m`)
- 1 Word file (`Lab06_YourLastName.docx`); use the template provided on Moodle. Your image files will be included in the Word document and should not be submitted separately.