Lab 2: Leslie Matrices

`CAS CS 132`: *Geometric Algorithms*

Due October 2, 2025 by 8:00PM

In this lab, you will be using NumPy and Matplotlib to run some experiments related to population dynamics. You're required to submit a write-up for this lab as a pdf on Gradescope. The details of what you're required to submit are given in the last section called "Lab Write-up."

Introduction

The Fibonacci sequence is defined by the following recurrence relation:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \end{aligned}$$

In other words, every number in the sequence (except for the first two) is the sum of the previous two numbers in the sequence:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$$

You likely already know this. What you may not know is Leonardo Bonacci (a.k.a. Fibonacci) came to this sequence by thinking about population dynamics. Consider the following scenario.

> *Suppose the following about rabbits:*
>
> - *A rabbit is either a juvenile or an adult. A rabbit is born as a juvenile, and after one month becomes (and remains) an adult.*
> - *A pair of adult rabbits breed and produce a pair of juvenile rabbits once every month.*
> - *Rabbits never die.*
>
> *Suppose that we begin with a pair of juvenile rabbits. How many pairs of adult rabbits do we have after* **n** *months?*

This is what the Fibonacci sequence tells us.

- After 1 month, we have 0 pairs of juvenile rabbits and 1 pair of adult rabbits (the juveniles who aged up).
- After 2 months, we have 1 pair of juvenile rabbits (born of the single pair of adults) and 1 pair of adult rabbits (the same as last month).

- After 3 months, we have 1 pair of juvenile rabbits (born of the single pair of adults) and 2 pairs of adult rabbit (the same as last month, plus the juveniles who became adults).
- After n months, we have F_{n-1} pairs of juvenile rabbits (born of the F_{n-1} pairs of adults from last month) and F_n pairs of adult rabbits (the F_{n-1} pairs of adults from last month, plus the F_{n-2} pair of juveniles who became adults).

The number of juveniles and adults after n months is linear in the number of juveniles and adults after $n-1$ months. Formally speaking, let J_n denote the number of pairs of juveniles after n months and A_n be denote the number of pairs of adults after n months. It follows that

- J_n is A_{n-1} (every pair of adults produces a pair of juveniles);
- A_n is $A_{n-1} + J_{n-1}$ (same adults as last month, plus the juveniles who became adults).

In symbols:

$$\begin{aligned} J_n &= A_{n-1} \\ A_n &= A_{n-1} + J_{n-1} \end{aligned}$$

or, in terms of matrix-vector multiplication:

$$\begin{bmatrix} J_{n+1} \\ A_{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} J_n \\ A_n \end{bmatrix}$$

and since $A_n = F_n$, and $J_n = A_{n-1} = F_{n-1}$, we also have a matrix-vector-multiplication-form for the Fibonacci recurrence:

$$\begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_{n-2} \\ F_{n-1} \end{bmatrix}$$

In particular, we can determine the nth Fibonacci number by repeated matrix-vector multiplications:

$$\begin{aligned}
\begin{bmatrix} F_3 \\ F_4 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_2 \\ F_3 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \right) \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_0 \\ F_1 \end{bmatrix} \right) \right) \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right) \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} 2 \\ 3 \end{bmatrix}
\end{aligned}$$

Leslie matrices are a generalization of the matrix form of the Fibonacci recurrence, using a more realistic scenario for modeling population dynamics.

> *Suppose the following about rabbits:*
>
> - *Only an s_i percentage of rabbits survive from i months old to $(i+1)$ months old. In particular, only s_0 percent of rabbits born survive to 1 month old.*
> - *For some fixed parameter N, only an s_N percentage of rabbits that are N months old or older survive each month.*
> - *Every rabbit that is i months old produces m_i rabbits on average each month.*
> - *For some fixed parameter N, every rabbit that is N months old or older produces m_N rabbits on average each month.*
>
> *Suppose we begin with k_i rabbits that are i months old for $i = 1, \ldots, N-1$ and k_N rabbits that are N months old or older. What is the total population after* **n** *months?*

Part of your task will be to determine how to construct a matrix given the parameters outlined in the above scenario. To help you with this, let's look at how this applies to Fibonacci's rabbits.

In Fibonacci's scenario, we only cared about the distinction between juveniles (1 month old) and adults ($\geq$ 2 month old), so we can take our parameter N to be 2. We know that 100% of all rabbits that are *i* months old survive, and that every adult rabbit produces 1 rabbit on average (a pair produced a pair), and every juvenile rabbit produced 0 rabbits on average. All together, we have the following parameters:

$$\begin{aligned} s_0 &= 1 \\ s_1 &= 1 \\ s_2 &= 1 \\ m_1 &= 0 \\ m_2 &= 1 \end{aligned}$$

In terms of J_n and A_n, this gives us:

$$\begin{aligned} J_n &= s_0 m_1 J_{n-1} + s_0 m_2 A_{n-1} = A_{n-1} \\ A_n &= s_1 J_{n-1} + s_2 A_{n-1} = A_{n-1} + J_{n-1} \end{aligned}$$

which gives us the recurrence relation above. So the Fibonacci matrix is a particular case of a Leslie matrix.

Next suppose we that only 80% of juveniles make it to adulthood, and only 70% of adults make it through the next month, but all rabbits born survive to become juveniles. Also, suppose that only 75% percent of adult breeding is successful. Then our parameters become:

$$\begin{aligned} s_0 &= 1 \\ s_1 &= 0.8 \\ s_2 &= 0.7 \\ m_1 &= 0 \\ m_2 &= 0.75 \end{aligned}$$

and in terms of J_n and A_n:

$$J_n = s_0 m_1 J_{n-1} + s_0 m_2 A_{n-1} = (0.75) A_{n-1}$$
$$A_n = s_1 J_{n-1} + s_2 A_{n-1} = (0.8) J_{n-1} + (0.7) A_{n-1}$$

and in terms of matrix-vector multiplication:

$$\begin{bmatrix} J_n \\ A_n \end{bmatrix} = \begin{bmatrix} 0 & 0.75 \\ 0.8 & 0.7 \end{bmatrix} \begin{bmatrix} J_{n-1} \\ A_{n-1} \end{bmatrix}$$

Thus, we can determine the total population after n months by calculating the vector $\begin{bmatrix} J_n \\ A_n \end{bmatrix}$ using repeated matrix-vector multiplication and then adding together its entries $J_n + A_n$.

Generally speaking the Leslie matrix is the matrix $L \in \mathbb{R}^{N \times N}$ such that, given a vector

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix}$$

where p_i is the number of rabbits that are i months old and p_N is the number of rabbits that are $\geq N$ months old, the vector $L\mathbf{p}$ is the new population after 1 month, i.e., the ith entry of $L\mathbf{p}$ is the number of rabbits that are i months old, after 1 month has passed, starting from the population given by $\mathbf{p}$.

Matplotlib

Before going onto describe the task for this lab, much of what you will be doing is building plots that show the population of rabbits over time, given a collection of parameters. During lab, you will go over with our Teaching Fellows how to make plots using Matplotlib. On your own you should read the Pyplot tutorial,[1] primarily the section titled "Formatting the style of your plot." **We will not guarantee that we will help you with basic Pyplot usage in office hours.** In particular, you must attend your discussion section to see this material.

Lab Write-up

The items in **bold** are what must be included in your write-up.

1. Implement the function `leslie_matrix` which constructs the Leslie matrix given survival parameters (s_i) and maternity parameters (m_i). See the docstring in the starter code for more details.

   Include your implementation in your write-up. Please do not include the docstring, just the code you wrote.

2. Implement the function `estimate_populations` which estimates the population of a rabbits each month using a Leslie matrix. See the docstring in the starter code for more details.

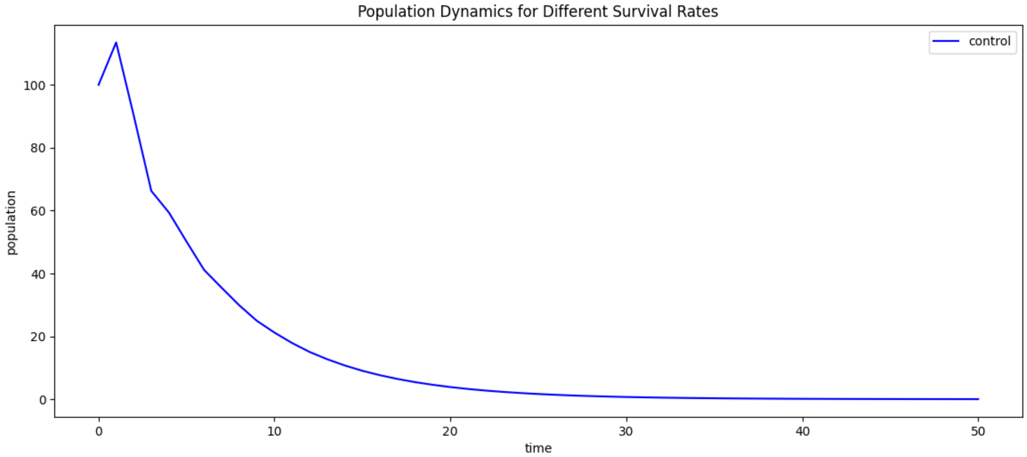
   Include your implementation in your write-up. Please do not include the docstring, just the code you wrote.

[1] https://matplotlib.org/stable/tutorials/pyplot.html#sphx-glr-tutorials-pyplot-py

3. Suppose you're given the following data for a population of rabbits.

i	0	1	2	3	4	5	6	7
initial pop. (p)	0	10	20	30	30	10	0	0
survival prob. (s)	0.5	0.5	0.6	0.7	0.6	0.5	0.3	0.3
maternity (m)	0	0	0.5	2	1	0.5	0.5	0

If you graph the population according the Leslie matrix for $N = 50$ months, you should get the following:



Describe in one to two sentences what this graph tells you about the population in the long term.

4. Suppose that you're interested in ensuring the survival of the population, and by conservation efforts, you are able to increase the survival rate of rabbits that are 4 months old. In the best case, this would mean increasing s_4 to be 1. Use the Python functions above to recreate the graph from the previous part, but with $s_4 = 1$.

   Describe in one to two sentences what this graph tells you about the population in the long term, assuming $s_4 = 1$. You do not need to include the graph for the part (but you will need to include it in a later part).

5. **Repeat the previous part but with $s_3 = 1$, $s_2 = 1$, and $s_1 = 1$. This means looking at three separate graphs; you should update one number at a time (i.e., don't consider the case in which the survival probabilities are 1 simultaneously).**

6. When conserving a population, the best-case scenario is that the population stabilizes; we don't want under-population or over-population. Using guess-and-check, estimate the value to 3 decimal places that you can assign to s_1 so that the population stabilizes.

   Include this value in your write-up.

7. Create a graph using Pyplot containing *all* plots from the previous parts on the same graph. This should be 6 plots in total. Your graph should have a title, axis labels and a legend. Furthermore, you should restrict the x-axis to be 0 to 50 and the y-axis to be 0 to 200 (see the starter code for some hints).

   Include the graph in your write-up.