# Eigenvalues in Simple Population and Markov Models

Golam Kabid

Department of Mathematics

Math 3030 – Mathematical Inquiry II

gkabid@mun.ca

February 2, 2026

**Abstract**

Eigenvalues are fundamental tools for analyzing the long-term behavior of linear systems. This report examines how eigenvalues can be used to study discrete population models and simple Markov chains. By combining theoretical analysis with numerical examples, we show how dominant eigenvalues determine whether a system grows, declines, or converges to a steady-state distribution.

## 1 Introduction

Many real-world systems evolve over time in a step-by-step manner. Examples include population growth, migration between regions, and random processes such as board games or weather patterns. These systems are often modeled using matrices that describe how the current state influences the next state.

Linear algebra provides powerful tools for studying such systems. In particular, eigenvalues offer insight into the long-term behavior of repeated matrix applications. Rather than computing large matrix powers directly, eigenvalues allow us to predict whether a system stabilizes, grows without bound, or decays over time.

Population models and Markov chains are two important applications of these ideas. Population models describe how groups change in size, while Markov chains model random transitions between states. Both rely on matrix multiplication, making eigenvalues a natural tool for analysis.

## 2 Methods

### 2.1 Discrete Linear Systems

A discrete-time linear system is defined by

$$\mathbf{x}_{n+1} = A\mathbf{x}_n, \tag{1}$$

where $\mathbf{x}_n$ represents the state at time $n$, and $A$ is a fixed matrix.

An eigenvalue $\lambda$ of $A$ satisfies

$$A\mathbf{v} = \lambda\mathbf{v}, \tag{2}$$

for some nonzero vector $\mathbf{v}$. Eigenvalues describe how vectors are scaled by the matrix, and the eigenvalue with the largest magnitude typically governs long-term behavior.

## 2.2 Eigenvalue Computation

Eigenvalues are computed by solving the characteristic equation

$$\det(A - \lambda I) = 0. \tag{3}$$

This method is applied to both population and Markov matrices to determine growth rates and steady states.

## 2.3 Numerical Simulation

To verify theoretical predictions, matrices are repeatedly applied to initial vectors. The numerical results are then compared with predictions made using eigenvalues.

# 3 Results

## 3.1 Population Model

Consider a population divided into two interacting groups:

$$\mathbf{x}_{n+1} = \begin{pmatrix} 1.1 & 0.2 \\ 0.1 & 0.9 \end{pmatrix} \mathbf{x}_n. \tag{4}$$

Solving the characteristic equation yields approximate eigenvalues

$$\lambda_1 \approx 1.2, \quad \lambda_2 \approx 0.8. \tag{5}$$

Since the dominant eigenvalue is greater than one, the model predicts population growth.

Starting with $\mathbf{x}_0 = (100, 50)^T$, repeated iteration gives the results in Table 1.

| Time $n$ | Group 1 | Group 2 |
|:---:|:---:|:---:|
| 0 | 100 | 50 |
| 1 | 120 | 55 |
| 2 | 143 | 63 |
| 3 | 170 | 72 |

Table 1: Population growth over time

## 3.2 Markov Chain Model

A Markov chain is defined by

$$\mathbf{x}_{n+1} = P\mathbf{x}_n, \tag{6}$$

where $P$ is a stochastic matrix. Consider

$$P = \begin{pmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{pmatrix}. \tag{7}$$

For any stochastic matrix, $\lambda = 1$ is an eigenvalue, and all other eigenvalues satisfy $|\lambda| < 1$. This guarantees convergence to a steady-state distribution.

Starting from $\mathbf{x}_0 = (1,0)^T$, the system evolves as shown in Table 2.

| Time $n$ | State 1 | State 2 |
|:---:|:---:|:---:|
| 0 | 1.00 | 0.00 |
| 1 | 0.80 | 0.20 |
| 2 | 0.70 | 0.30 |
| 3 | 0.65 | 0.35 |

Table 2: Convergence of a Markov chain

# 4 Conclusion

This report demonstrates how eigenvalues provide a clear and efficient method for analyzing the long-term behavior of linear systems. In population models, eigenvalues predict growth or decline, while in Markov chains they ensure convergence to a steady-state distribution. Future work could extend these ideas to larger systems, nonlinear models, or real-world data.

## Acknowledgments

The author thanks the instructor of Math 3030 for guidance during the project.

# A Sample Computational Code

A simple algorithm for iterating a population model:

```
x = [100, 50]
for n in range(10):
    x = A @ x
    print(x)
```

# References

[1] D. Lay, S. Lay, and J. McDonald, *Linear Algebra and Its Applications*, Pearson Education.

[2] G. Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press.

[3] C. Grinstead and J. Snell, *Introduction to Probability*, American Mathematical Society.