

Producing the Fibonacci Numbers by Repeated Iteration of a Matrix

Tyler Murphy

Boise State University

18 Dec 2013

Outline

Introduction

Motivation

Key Terminology

Some Graphical Examples of Iterations

Markov

Matrix of Det less than 1

Matrix of Det less than 1

Fibonacci Application

Constructing the Fibonacci Sequence through Matrix iteration

conclusion

Motivation

- ▶ Matrices are simple, effective ways to represent a system of equations. However, sometimes we are not interested in what happens in a system in the short term. We want to analyze trends and see where things are going.
- ▶ To this end, we want to see what happens after the system repeats itself over time.

For our purposes we will be looking at the repeated iteration of a 2x2 matrix:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

Key Terminology

- ▶ **Determinant** = the difference in the product of the two diagonals.
- ▶ **Iteration** = Repetition of a process.
- ▶ **Eigenvectors**, v are vectors such that when multiplied by a matrix A , they yield a multiple of that vector. That multiple is called the **eigenvalue**, λ .

$$Av = \lambda v$$

Graphical Examples

- ▶ A Markov matrix is a matrix whose columns add up to 1.
- ▶ Consider matrix $A =$

$$\begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix}$$

under iteration for 10,000 pairs of randomly chosen initial conditions.

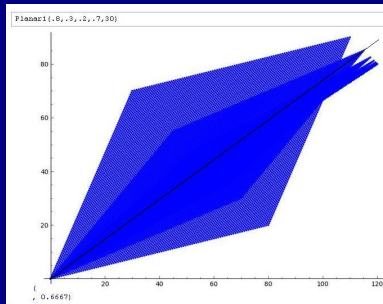


Figure 1 : Repeated iterates (x_n, y_n) of a markov matrix A

Graphical examples

- ▶ Now we investigate a matrix with a positive determinant < 1 .
- ▶ Consider matrix $A =$

$$\begin{bmatrix} 1 & \frac{1}{10} \\ \frac{99}{100} & 1 \end{bmatrix}$$

under iteration for 10,000 pairs of randomly chosen initial conditions.

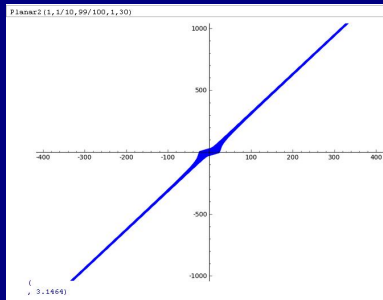


Figure 2 : Repeated iterates (x_n, y_n) of a matrix A

Graphical examples

- ▶ Now we investigate a matrix with a positive determinant > 1 .
- ▶ Consider matrix $A =$

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

under iteration for 10,000 pairs of randomly chosen initial conditions.

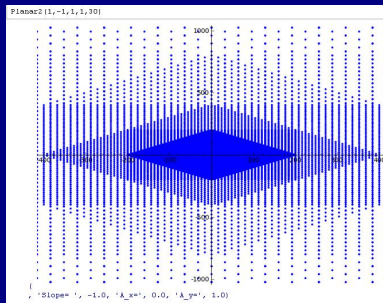


Figure 3 : Repeated iterates (x_n, y_n) of a matrix A

Graphical Examples

- ▶ Now consider a matrix $A =$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

under iteration of initial values $[1,0]$

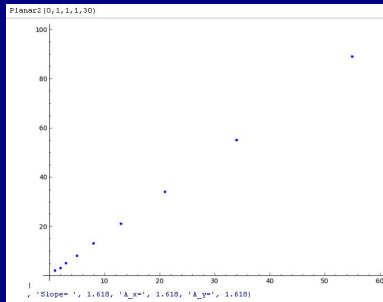


Figure 4 : Repeated iterates of a matrix A beginning at $[1,0]$

We see that this matrix generates the fibonacci sequence. But why?

Fibonacci Sequence

- ▶ The fibonacci Sequence is something we are all familiar with.
- ▶ 1,1,2,3,5,8,13,21,34 ...
- ▶ Each term is the sum of the previous two terms, with the first two terms being 1.
- ▶ We can generalize this by saying that $F_{n+1} = F_n + F_{n-1}$.
- ▶ But what if you want to find the n^{th} term? It would be very tedious to have to generate the entire sequence just to get where you want to be.
- ▶ To that end, we can use the iteration of a matrix and the fact that this matrix is diagonalizable.

Fibonacci Generation

First consider how this matrix was generated. A matrix is just a simple way of writing a system of equations.

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} F_n & + & F_{n+1} \\ F_n & & \end{bmatrix}$$

Which we can translate to:

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

Fibonacci Generation

- ▶ Let's test a few examples to make sure this is working properly.
- ▶ Consider:

$$\begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

▶

$$\begin{bmatrix} F_2 \\ F_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

▶

$$\begin{bmatrix} F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

▶

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Finding the Eigenvalues

- ▶ Since computing each power of the matrix can be time consuming and lengthy, we first diagonalize the matrix with the use of its *eigenvectors* and *eigenvalues*
- ▶ First we compute the eigenvalues, λ , of the matrix in the following manner:

$$\begin{vmatrix} 1 - \lambda & 1 \\ 1 & 0 - \lambda \end{vmatrix} = (1 - \lambda)(-\lambda) - 1 = 0. \quad (1)$$

$$\lambda^2 - \lambda - 1 = 0 \quad (2)$$

Solving (2) we get:

$$\lambda = \frac{1 \pm \sqrt{1 - 4(1)(-1)}}{2(1)} = \frac{1 \pm \sqrt{5}}{2} \quad (3)$$

Finding the Eigenvectors

Now we can use the eigenvalues to solve for the eigenvectors.

$$\begin{bmatrix} 1 - \frac{1+\sqrt{5}}{2} & 1 \\ 1 & -\frac{1+\sqrt{5}}{2} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1-\sqrt{5}}{2} & 1 \\ 1 & -\frac{1+\sqrt{5}}{2} \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1 & -\frac{1+\sqrt{5}}{2} \\ \frac{1-\sqrt{5}}{2} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -\frac{1+\sqrt{5}}{2} \\ 0 & 0 \end{bmatrix}$$

Which gives us, for λ_1 , $\vec{x}_1 =$

$$\begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{bmatrix}$$

Repeating this process for λ_2 , we get $\vec{x}_2 =$

$$\begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{bmatrix}$$

Diagonalizing $A = SDS^{-1}$

We can now place \vec{x}_1, \vec{x}_2 into a matrix, $S =$

$$\begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix}$$

Matrix D is the diagonal matrix whose entries are the eigenvalues whose position corresponds to the column of their corresponding eigenvector in S . So $D =$

$$\begin{bmatrix} \frac{1+\sqrt{5}}{2} & 0 \\ 0 & \frac{1-\sqrt{5}}{2} \end{bmatrix}$$

Diagonalizing $A = SDS^{-1}$

Now recall that

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = A^n \begin{bmatrix} 1 \\ 0 \end{bmatrix} = (SDS^{-1})^n \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (4)$$

$$= SD^n S^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (5)$$

Computing the Final Formula

Now we substitute our values for S , D , and S^{-1} .

$$\begin{aligned} & \begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \\ & \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \left(\frac{1+\sqrt{5}}{2}\right)^n & 0 \\ 0 & \left(\frac{1-\sqrt{5}}{2}\right)^n \end{bmatrix} \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ & = \frac{1}{\sqrt{5}} \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \left(\frac{1+\sqrt{5}}{2}\right)^n & 0 \\ 0 & \left(\frac{1-\sqrt{5}}{2}\right)^n \end{bmatrix} \begin{bmatrix} 1 & -\left(\frac{1-\sqrt{5}}{2}\right) \\ -1 & \frac{1+\sqrt{5}}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{aligned}$$

Finalizing the Fibonacci formula

Computing the matrix multiplication we get:

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} \left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1} \\ \left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \end{bmatrix} \quad (6)$$

And so,

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right] \quad (7)$$

Conclusion

- ▶ As we have seen, Matrix iteration has lots of interesting and practical applications. Of the practical side specifically are Markov matrices which are used in modeling population migration, manufacturing, economic trends, and many other areas where there is a finite supply involved and trends are sought.
- ▶ Matrix iteration also has application in the mathematical world in helping to create formulas to find specific values in a sequence, like the Fibonacci and Gibonacci numbers.

Further Study

- ▶ Examine limiting trends of matrices under iteration given a randomly chosen initializing vector.
- ▶ What is the slope of the line of convergence of a matrix? How does this relate to the eigenvectors and eigenvalues?
- ▶ Under what condition does a matrix fail to converge?
- ▶ What initial values cause a matrix to stay fixed?