Difference Equations

Math 45 — Linear Algebra

David Arnold

David-Arnold@Eureka.redwoods.cc.ca.us

Abstract

This activity investigates the form of closed form solutions of first order difference equations of the form $\mathbf{u}_k = A\mathbf{u}_{k-1}$ where A is a 2×2 matrix. *Prerequisites: This activity requires a knowledge of eigenvalues and eigenvectors.*

next page

close

exit

Table of Contents

title page
contents
previous page
next page
back
print doc
close
exit

First Order Difference Equations

What follows is called a *first order difference equation* with initial condition.

$$a_n = 1.2a_{n-1}, \quad a_0 = 2 \tag{1}$$

You might find this form of equation familiar, particularly if you have studied recursively defined sequences in a college algebra class. The equation and initial condition in **Equation 1** are easily used to produce the following sequence of numbers.

$$\begin{aligned} a_1 &= 1.2a_0 = 1.2(2) \\ a_2 &= 1.2a_1 = (1.2)^2(2) \\ a_3 &= 1.2a_2 = (1.2)^3(2) \\ &\vdots \end{aligned} \tag{2}$$

The sequence of equations in **Equation 2** show that the nth term of the sequence generated by **Equation 1** is given by $a_n = (1.2)^n(2)$ or $a_n = 2(1.2)^n$. The equation $a_n = 2(1.2)^n$ is called the *closed form solution* of **Equation 1**. The closed form solution can easily be used to find the 10th term of the sequence generated by **Equation 1**.

$$\begin{aligned} a_{10} &= 2(1.2)^{10} \\ a_{10} &\approx 12.3835 \end{aligned}$$

Using Matlab

Let's use Matlab to produce the first 11 terms of the sequence generated by the first order difference **Equation 1**. First, declare an array of zeros which will be used to store 11 terms of the sequence. In **Equation 1**, note that the value of the first term is $a_0 = 2$. Set this entry in the first position of vector **a**.

title page
contents
previous page
next page
back
print doc
close
exit

```
>> a=zeros(11,1);
>> a(1)=2
a =
     2
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
```

According to **Equation 1**, we generate the k th term by multiplying the $k-1$ st term by 1.2. That is, each term in the sequence is generated by multiplying the previous term by 1.2. This is easy to do in Matlab if we use a `for` loop.

```
>> for k=2:11,a(k)=1.2*a(k-1);,end
>> a
a =
    2.0000
    2.4000
    2.8800
    3.4560
    4.1472
    4.9766
    5.9720
    7.1664
```

title page
contents
previous page
next page
back
print doc
close
exit

```
      8.5996
     10.3196
     12.3835
```

This bit of code bears some explaining. The first entry of the vector **a** already contains the initial condition $a_0 = 2$. Because the next entry will be stored in the second component of the vector **a**, we begin our loop with $k = 2$. The notation `2:11` produces a vector that generates a vector, starting with 2, incrementing by 1 (the default), and ending with 11. Thus, the first time through the loop, $k = 2$, and each time we iterate, k is incremented by 1 until the last time we pass through the loop, when $k = 11$. Each time we iterate, the kth entry of the vector is computed and stored in `a(k)`. The command `end` signals the end of the `for` loop.

Matrices and Difference Equations

First order difference equations involving matrices and vectors can also have closed form solutions. For example, consider the following first order difference equation with initial condition.

$$\mathbf{x}_k = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x}_{k-1}, \quad \mathbf{x}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3}$$

The difference **Equation 3** can be used to produce a sequence of *vectors* in a manner similar to the way we generated a sequence of numbers with **Eequation 1**.

$$\begin{aligned}
\mathbf{x}_1 &= \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x}_0 = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
\mathbf{x}_2 &= \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x}_1 = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \\
\mathbf{x}_3 &= \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x}_2 = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 7 \\ 8 \end{pmatrix} \\
&\vdots
\end{aligned}$$

title page
contents
previous page
next page
back
print doc
close
exit

Using Matlab

You can use Matlab to produce a sequence of vectors generated by the difference **Equation 3**. First, enter the matrix A and the initial condition $\mathbf{x}_0$.

```
>> A=[1 1;0 2]
A =
     1     1
     0     2
>> x0=[0;1]
x0 =
     0
     1
```

Let's generate a sequence of 11 terms. This time, each term of the sequence is a 2×1 vector. Thus, reserve space in matrix X for 11 such vectors, each of which will be stored as a column in matrix X. The initial condition $\mathbf{x}_0 = (0, 1)^T$ is placed in the first column of matrix X.

```
>> X=zeros(2,11);
>> X(:,1)=x0
X =
     0     0     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0     0
```

Remember, the notation `X(:,1)` is read "every row, 1 st column. Thus, the command `X(:,1)=x0` stores the contents of the initial condition in the first column of the matrix X.

In a manner similar to our last example, the k th term of the sequence is calculated by multiplying the $k-1$ st term of the sequence by the matrix A. That is, each term of the sequence is generated by multiplying the previous term of the sequence by the matrix A. Again, we use a `for` loop.

title page
contents
previous page
next page
back
print doc
close
exit

```
>> for k=2:11,X(:,k)=A*X(:,k-1);end
>> X
X =
  Columns 1 through 6
           0           1           3           7          15          31
           1           2           4           8          16          32
  Columns 7 through 11
          63         127         255         511        1023
          64         128         256         512        1024
```

Again, the first time through the loop, $k = 2$. Thereafter, we iterate through the loop, incrementing by one, until we iterate a final time with $k = 11$. The body of the loop bears some explaining. The notation `X(:,k)` is read "every row, kth column," while the notation `X(:,k-1)` is read "every row, $k-1$st column. Thus, the command `X(:,k)=A*X(:,k-1)` multiplies the $k-1$st column of X by the matrix A, storing the result in the kth column of matrix X.

It is clear from this last computation that

$$\mathbf{x}_{10} = \begin{pmatrix} 1023 \\ 1024 \end{pmatrix}$$

Keep this result in mind for later comparisons.

Closed Form Solutions

We will now try to produce a closed form solution for the following difference equation with initial condition.

$$\mathbf{x}_k = A\mathbf{x}_{k-1}, \quad \mathbf{x}_0 = \mathbf{x}_0 \tag{4}$$

title page
contents
previous page
next page
back
print doc
close
exit

Suppose that the matrix A has eigenvalues λ_1 and λ_2 with associated eigenvectors[1] $\mathbf{v}_1$ and $\mathbf{v}_2$. Furthermore, suppose that the initial condition $\mathbf{x}_0$ can be expressed as a linear combination of the eigenvectors in the following manner[2].

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2$$

We can find $\mathbf{x}_1$ as follows:

$$\begin{aligned}\mathbf{x}_1 &= A\mathbf{x}_0 \\ &= A(c_1\mathbf{v}_1 + c_2\mathbf{v}_2) \\ &= c_1A\mathbf{v}_1 + c_2A\mathbf{v}_2 \\ &= c_1\lambda_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2\end{aligned}$$

We can find $\mathbf{x}_2$ as follows:

$$\begin{aligned}\mathbf{x}_2 &= A\mathbf{x}_1 \\ &= A(c_1\lambda_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2) \\ &= c_1\lambda_1A\mathbf{v}_1 + c_2\lambda_2A\mathbf{v}_2 \\ &= c_1\lambda_1\lambda_1\mathbf{v}_1 + c_2\lambda_2\lambda_2\mathbf{v}_2 \\ &= c_1\lambda_1^2\mathbf{v}_1 + c_2\lambda_2^2\mathbf{v}_2\end{aligned}$$

In a similar manner, $\mathbf{x}_3 = c_1\lambda_1^3\mathbf{v}_1 + c_2\lambda_2^3\mathbf{v}_2$. If you continue in this manner, it will be evident that the closed form solution of difference **Equation 4** is given by the following equation.

$$\mathbf{x}_k = c_1\lambda_1^k\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 \tag{5}$$

---

[1] Recall that a non-zero solution $\mathbf{x}$ of $A\mathbf{x} = \lambda\mathbf{x}$ is called an eigenvector and λ is its associated eigenvalue.

[2] This can always be done if matrix A is diagonalizable; that is, if the $n \times n$ matrix A has n linearly independent eigenvectors.

title page
contents
previous page
next page
back
print doc
close
exit

Using the Theory

Theory is great, but only if it produces correct results. Let's use **Equation 5** to find the solution of the difference **Equation 3**. First, let's repeat **Equation 3** here so we won't have to do too much page turning every time we wish to reference **Equation 3**.

$$\mathbf{x}_k = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x}_{k-1}, \quad \mathbf{x}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6}$$

To find a closed form solution of **Equation 6**, proceed as follows:

1. Find the eigenvalues and eigenvectors of the matrix $A = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$.
2. Express the initial condition as a linear combination of the eigenvectors.
3. Use **Equation 5** to write the closed form solution and test the result.

The characteristic equation of matrix A is $p(\lambda) = \lambda^2 - 3\lambda + 2$. The eigenvalues, the roots of the characteristic polynomial, are $\lambda_1 = 1$ and $\lambda_2 = 2$. The following command computes the characteristic polynomial of matrix A.

```
>> p=poly(A)
p =
     1    -3     2
```

Note that the coefficients here are in descending powers of λ. Thus, `[1 -3 2]` represents the polynomial $p(\lambda) = \lambda^2 - 3\lambda + 2$. The next command computes the roots of the characteristic polynomial, which are, of course, the eigenvalues of the matrix A.

```
>> roots(p)
ans =
```

title page
contents
previous page
next page
back
print doc
close
exit

```
       2
       1
```

The eigenspace for each eigenvalue λ is captured by finding the nullspace of $A - \lambda I$. Although this is easy enough to do by hand (and you should practice this skill), let's use the Matlab `null` command to produce an eigenvector for each eigenvalue.

```
>> v1=null(A-1*eye(2),'r')
v1 =
       1
       0
>> v2=null(A-2*eye(2),'r')
v2 =
       1
       1
```

The `'r'` switch causes Matlab to compute the eigenvector in a manner similar to the technique you would use to compute the eigenvector with pencil and paper calculations. If you do not use the `'r'` switch, Matlab computes an orthonormal basis for the eigenspace. The eigenvectors associated with $\lambda_1 = 1$ and $\lambda_2 = 2$ are $\mathbf{v}_1 = (1,0)^T$ and $\mathbf{v}_2 = (1,1)^T$, respectively.

Our second task is to write $\mathbf{x}_0$ as a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$.

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} = c_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

This vector equation can be written as a matrix equation.

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Note that this last equation has the form $V\mathbf{c} = \mathbf{x}_0$, where

title page
contents
previous page
next page
back
print doc
close
exit

$$V = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \qquad \text{and} \qquad \mathbf{x}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The eigenvectors, now the columns of the coefficient matrix V, are independent. Thus, the coefficient matrix V is invertible, and the solution of $V\mathbf{c} = \mathbf{x}_0$ is $\mathbf{c} = V^{-1}\mathbf{x}_0$. This is easily solved with Matlab. First, construct the matrix V. Note that the columns of V are the eigenvectors; i.e., $V = [\mathbf{v}_1, \mathbf{v}_2]$.

```
>> V=[v1,v2]
V =
     1     1
     0     1
```

Load the initial condition.

```
>> x0=[0;1]
x0 =
     0
     1
```

Calculate **c**.

```
>> c=inv(V)*x0
c =
    -1
     1
```

Hence,

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

title page
contents
previous page
next page
back
print doc
close
exit

Finally, substitute the constants, eigenvalues, and eigenvectors into **Equation 5**.

$$
\begin{aligned}
\mathbf{x}_k &= c_1\lambda_1^k\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 \\
\mathbf{x}_k &= (-1)(1)^k\begin{pmatrix}1\\0\end{pmatrix} + (1)(2)^k\begin{pmatrix}1\\1\end{pmatrix}
\end{aligned}
$$

Simplifying,

$$\mathbf{x}_k = \begin{pmatrix}-1\\0\end{pmatrix} + 2^k\begin{pmatrix}1\\1\end{pmatrix}. \tag{7}$$

You should always check your work. For example, to find $\mathbf{x}_{10}$ substitute $k = 10$ in **Equation 7**.

$$
\begin{aligned}
\mathbf{x}_{10} &= \begin{pmatrix}-1\\0\end{pmatrix} + 2^{10}\begin{pmatrix}1\\1\end{pmatrix} \\
\mathbf{x}_{10} &= \begin{pmatrix}-1\\0\end{pmatrix} + \begin{pmatrix}1024\\1024\end{pmatrix} \\
\mathbf{x}_{10} &= \begin{pmatrix}1023\\1024\end{pmatrix}
\end{aligned}
$$

Note that this is in agreement with the solution found earlier. You can also use Matlab and **Equation 7** to produce a number of terms generated by the first order difference **Equation 3**.

```
>> X=zeros(2,11);
>> for k=1:11,X(:,k)=[-1;0]+2^(k-1)*[1;1];end
>> X
X =
   Columns 1 through 6
            0            1            3            7           15           31
            1            2            4            8           16           32
   Columns 7 through 11
```

title page
contents
previous page
next page
back
print doc
close
exit

```
      63        127        255        511        1023
      64        128        256        512        1024
```

Note that this sequence of numbers is in complete agreement with those found by our earlier iteration of **Equation 3**.

Homework

Perform each of the following tasks for each of the following first order difference equations.

- Use Matlab to produce the first 11 terms of the sequence generated iteratively by the difference equation.
- Use Matlab's `poly` and `roots` commands to find the characteristic polynomial and eigenvalues.
- Use the Matlab's `null` to find the associated eigenvectors for each eigenvalue.
- Express the initial condition as a linear combination of the eigenvectors.
- Write the closed form solution of the difference equation.
- Use Matlab and the closed form solution to produce the first 11 terms of the sequence generated by the difference equation.

1. $\mathbf{x}_k = \begin{pmatrix} -3 & 0 \\ 5 & 2 \end{pmatrix} \mathbf{x}_{k-1}, \quad \mathbf{x}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

2. $\mathbf{x}_k = \begin{pmatrix} 2.5 & 0.5 \\ 0.5 & 2.5 \end{pmatrix} \mathbf{x}_{k-1}, \quad \mathbf{x}_0 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$

title page
contents
previous page
next page
back
print doc
close
exit

Using hand calculations only, find a closed form solution of the following first order difference equation.

3. $\mathbf{x}_k = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x}_{k-1}, \quad \mathbf{x}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

title page
contents
previous page
next page
back
print doc
close
exit