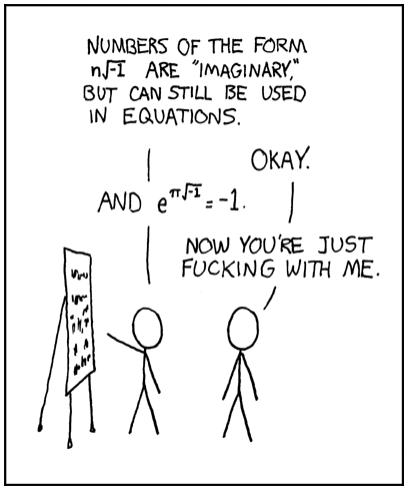
Quantitative Understanding in Biology Module III: Linear Difference Equations Lecture II: Complex Eigenvalues

Introduction

In the previous section, we considered the generalized two-variable system of linear difference equations, and showed that the eigenvalues of these systems are given by the equation…

$$\lambda_{1,2} = \frac{b \pm \sqrt{b^2 - 4c}}{2}$$

…where b = ($a_{11} + a_{22}$), c=($a_{22} \cdot a_{11} - a_{12} \cdot a_{21}$), and $a_{i,j}$ are the elements of the 2 x 2 matrix that define the linear system. Inspecting this relation shows that it is possible for the eigenvalues of such a system to be complex numbers. Because our plants and seeds model was our first look at these systems, it was carefully constructed to avoid this complication [exercise: can you show that this model cannot produce complex eigenvalues]. However, many systems of biological interest do have complex eigenvalues, so it is important that we understand how to deal with and interpret them. We'll begin with a review of the basic algebra of complex numbers, and then consider their meaning as eigenvalues of dynamical systems.



http://www.xkcd.org/179

A Review of Complex Numbers

You may recall that complex numbers can be represented with the notation a+b*i*, where a is the real part of the complex number, and b is the imaginary part. The symbol *i* denotes $\sqrt{-1}$ (recall i^2 = -1, i^3 = -*i* and i^4 = +1). Hence, complex numbers can be thought of as points on a complex plane, which has real and imaginary axes. In some disciplines, such as electrical engineering, you may see $\sqrt{-1}$ represented by the symbol *j*.

This geometric model of a complex number suggests an alternate, but equivalent, representation. Just as we can transform any point in Cartesian space into polar coordinates, we can choose to represent complex numbers as a distance, r, from 0+0i, and an angle, ϕ, from the real axis. Basic trigonometry gives us…

$$a = r\cos\phi \quad r = \sqrt{a^2 + b^2}$$
$$b = r\sin\phi \quad \phi = \tan^{-1}\frac{b}{a}$$

When writing complex numbers in this polar form, we use this notation: $re^{i\phi}$. This is not an arbitrary representation, and can be derived by considering the infinite series representations of sin x, cos x, and e^x. You may recall the series for the basic trigonometric functions:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \cdots$$

Now, our geometric interpretation of a complex number implies …

$$a + bi = r[\cos\phi + i\sin\phi\,]$$

…and the series expansions of sin x and cos x allow us to take this one step further…

$$a + bi = r\left[1 + \phi i - \frac{\phi^2}{2!} - \frac{\phi^3 i}{3!} + \frac{\phi^4}{4!} + \frac{\phi^5 i}{5!} - \frac{\phi^6}{6!} - \frac{\phi^7 i}{7!} + \frac{\phi^8}{8!} + \frac{\phi^9 i}{9!} - \cdots\right]$$

Now let us consider the series expansion of e^x. Recall…

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \frac{x^8}{8!} + \frac{x^9}{9!} + \cdots$$

Now if we allow x to take on an imaginary value, specifically, ϕ*i*, we get…

$$e^{\phi i} = 1 + \phi i + \frac{(\phi i)^2}{2!} + \frac{(\phi i)^3}{3!} + \frac{(\phi i)^4}{4!} + \frac{(\phi i)^5}{5!} + \frac{(\phi i)^6}{6!} + \frac{(\phi i)^7}{7!} + \frac{(\phi i)^8}{8!} + \frac{(\phi i)^9}{9!} + \cdots$$

$$e^{\phi i} = 1 + \phi i - \frac{\phi^2}{2!} - \frac{\phi^3 i}{3!} + + \frac{\phi^4}{4!} + \frac{\phi^5 i}{5!} - \frac{\phi^6}{6!} - \frac{\phi^7 i}{7!} + \frac{\phi^8}{8!} + \frac{\phi^9 i}{9!} - \cdots = \cos\phi + i\sin\phi$$

…or more generally

$$re^{\phi i} = r[\cos\phi + i\sin\phi]$$

One of the main motivations for this representation of complex numbers is that it gives us an intuitive understanding of what it means to raise a complex number to an integer power.

$$\left(re^{\phi i}\right)^n = r^n e^{n\phi i}$$

We see that when we raise a complex number to a power, its norm (or length) is scaled, and its angle is multiplied. This corresponds to a simultaneous scaling and rotation of the number in the complex plane.

Interpreting Complex Eigenvalues

Using the 'polar' representation of complex numbers just described, we are now in a position to interpret the meaning of a complex eigenvalue in a linear dynamical system. If we focus on the norm of

the eigenvalue, we see that raising a complex number to an arbitrarily large power will converge to 0+0i when the norm is less than one. The value will grow in an unbounded fashion if the norm is greater than unity, and the result will continue to rotate in the complex plane indefinitely with constant magnitude if the norm is exactly one.

If the angular part of the complex number is non-zero, the complex number will spiral around the origin as n increases. The larger the angle, the higher the frequency (and the smaller the period) of rotation. ***We therefore expect to see periodic behavior with a characteristic period or frequency in systems with complex eignevalues***.

Note that everything we know about real eignevalues are just degenerate cases of the complex eigenvalue for which ϕ = 0 for positive numbers and ϕ = 180° for negative numbers. You can think of ϕ=0 as the limiting case of an infinitely long period of rotation. Also, a negative real eigenvalue corresponds to a 180° rotation every step, which is simply alternating sign.

It is also worth noting that, because they ultimately come from a polynomial characteristic equation, complex eigenvalues always come in complex conjugate pairs. These pairs will always have the same norm and thus the same rate of growth or decay in a dynamical system. Also, they will be characterized by the same frequency of rotation; however, the directions of rotation will be opposing.

Note that our dynamical systems are defined solely by real numbers (it is not clear what a complex plant or seed count would mean). It turns out that the eigenvectors of a dynamical systems become complex when the eigenvalues are complex; this occurs in such a manner that we the imaginary parts disappear in the final values of the dynamical system's state variables.

Example: Consider a linear dynamical system with a matrix M = $\begin{bmatrix} 1.0005 & -0.05 \\ 0.05 & 1.0005 \end{bmatrix}$. We should get into the habit of trying to interpret model matrices when we see them, so before we go off and compute eigenvalues, let's see if we can't make sense of this matrix, even without any biological model behind it.

The governing equation for this model is…

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1.0005 & -0.05 \\ 0.05 & 1.0005 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

We see that, left to its own devices (i.e., if there were no interaction with y), x will grow in an unbounded fashion, but slowly with a λ just over 1.0. A similar statement can be made for y. However, there are off-diagonal terms in our model matrix, and we see that larger values of y lead to reductions in x. However, reductions in x will lead to reductions in y. Thus we might intuitively expect that this system has the potential to exhibit oscillatory behavior.

Now let us compute the eigenvalues of the model matrix.


```
>> M = [1.0005 -0.05; 0.05 1.0005]; ev = eig(M)

ev =

   1.0005 + 0.0500i
   1.0005 - 0.0500i
```

We see our complex conjugate eigenvalues, and observe that their norms are greater than 1. We can compute the angle of the first eigenvalue as follows:

```
>> angle(ev(1))
ans =
    0.0499
>> 2*pi/ans
ans =
  125.8311
```

So we expect a rotation every 126 steps or so. Let's 'simulate' our systems and see where it goes in the first 126 steps…

```
>> x = [50 50]; for (i=1:125) x(i+1,:) = M * x(i,:)'; end
>> plot(x)
>> plot(x(:,1), x(:,2))
```

We can, of course, simulate the system for longer periods of time…

```
>> x = [50 50]; for (i=1:1260) x(i+1,:) = M * x(i,:)'; end
>> plot(x(:,1), x(:,2))
>> plot(x)
```

Note that this model produces negative values for the variables. Fortunately, we did not specify what the model represents biologically, but in general one needs to think about the definitions of these models. Often physically realistic models will be prevented from producing non-negative values by the inclusion of non-linear terms. For example, if were modeling rabbits and foxes, the number of rabbits eaten by foxes might be modeled by an equation such as.

$$rabbits_{n+1} = \lambda\,(rabbits_n) - \alpha\,(rabbits_n)\,(foxes_n)$$

You can think of the second term as some fraction of the number of encounters between rabbits and foxes.

Consider the 3x3 model matrix M = $\begin{bmatrix} 0.8 & -0.05 & -0.05 \\ 0.05 & 1 & 0 \\ 0 & 0.05 & 1 \end{bmatrix}$. It is a bit harder to figure out what this model will do long term by analyzing the matrix, but it should be clear from the eigenvalues…


```
>> M = [0.8 -0.05 -0.05; 0.05 1.0 0; 0 0.05 1.0]; ev = eig(M)
ev =
   0.8097
   0.9952 + 0.0252i
   0.9952 - 0.0252i
>> norm(ev(2))
ans =
    0.9955
```

How do you think this system will behave?

```
>> x = [1 1 1]; for (i=1:1000) x(i+1,:) = M * x(i,:)'; end
>> plot(x)
>> plot3(x(:,1), x(:,2), x(:,3))
```

Reverse Engineering a 2x2 Linear Dynamical System

We can use our knowledge of complex eigenvalues to reverse-engineer linear systems that have the properties that we want. Suppose that we want a system that decays 0.1% per step, and exhibits oscillatory behavior with a period of 30 steps. We know that we want the eigenvalues of this system to be $\lambda_{1,2} = 0.999e^{\pm\frac{2\pi i}{30}}$.

We could continue with this specific numerical example, but it is not so difficult to generalize the problem to engineering a system with a growth parameter D and a period P. We seek to find a matrix $M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$ such that its eigenvalues are given by

$$\lambda_{1,2} = De^{\pm\frac{2\pi i}{P}}$$

Knowing that complex eigenvalues come in conjugate pairs, we'll confine ourselves to only the first eigenvaue. We have

$$Re(\lambda_1) = D\cos\frac{2\pi}{P}$$

$$Im(\lambda_1) = D\sin\frac{2\pi}{P}$$

Recall that for a 2x2 system, we also know…

$$\lambda = \frac{Tr(M) \pm \sqrt{Tr^2(M) - 4\,Det(M)}}{2}$$

The real part of the solution comes from the first term in the relation above, so we can write…

$$Re(\lambda_1) = \frac{Tr(M)}{2} = \frac{m_{11} + m_{22}}{2} = D\cos\frac{2\pi}{P}$$

We can arbitrarily decide that $m_{11} = m_{22}$ and conclude…

$$m_{11} = m_{22} = D\cos\frac{2\pi}{P}$$

The imaginary part of the solution for λ comes from the second term, so we can write…

$$i\,Im(\lambda_1) = \frac{\sqrt{Tr^2(M) - 4\,Det(M)}}{2}$$

$$-4\,Im^2(\lambda_1) = Tr^2(M) - 4\,Det(M)$$

Substituting our knowledge of the trace of M into this relation (and rearranging a bit) gives…

$$4D^2\cos^2\frac{2\pi}{P} + 4\,Im^2(M) = 4\,Det(M)$$

$$D^2\cos^2\frac{2\pi}{P} + D^2\sin^2\frac{2\pi}{P} = D^2 = Det(M)$$

$$D^2 = m_{11}m_{12} - m_{12}m_{21}$$

$$D^2 = D^2\cos^2\frac{2\pi}{P} - m_{12}m_{21}$$

$$D^2(1 - \cos^2\frac{2\pi}{P}) = -m_{12}m_{21}$$

$$D^2\sin^2\frac{2\pi}{P} = -m_{12}m_{21}$$

We will again make an arbitrary decision, this time that $m_{12} = -m_{21}$, so we can finally write…

$$m_{21} = -m_{12} = D\sin\frac{2\pi}{P}$$

Our final matrix is then…

$$M = \begin{bmatrix} D\cos\frac{2\pi}{P} & -D\sin\frac{2\pi}{P} \\ D\sin\frac{2\pi}{P} & D\cos\frac{2\pi}{P} \end{bmatrix}$$

Thinking about non-dynamic linear systems

So far, we have considered linear dynamic models. However, much of the linear algebra we have used can be used to describe generalized transformation. A matrix can also be thought of as an arbitrary

transformation of inputs to outputs. Matrices can be used to describe the effects of a lens of a microscope on light or the net effect of an MR machine. Of course, not all processes are linear, so not all processes can be encoded exactly by a matrix (although with proper treatment these may be decent approximations).

Consider the matrix we just derived for an oscillating system. If you imagine that what goes into the matrix is not two state variables in a dynamic system, but rather a pair of (x,y) coordinates, it should be clear that this matrix rotates the coordinates around the origin when D = 1.0. This is a standard rotation matrix, and is probably worth memorizing.

In fact, you can feed this matrix more than just a single coordinate. The system below takes three points in space as input, and the output is those three points in space rotated around the origin by an angle θ.

$$\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2\end{bmatrix}$$

Example: Can you design a matrix that reports the average of three sensors.

Similarly, the system below scales or zooms these points in or out of the origin by a factor D:

$$\begin{bmatrix}D & 0 \\ 0 & D\end{bmatrix}\begin{bmatrix}x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2\end{bmatrix}$$

Scaling and translation are two basic geometric transformations; the last is translation. Expressing translation operations in matrix form is a bit tricky because, technically speaking, translation is an affine, not a linear, operation (recall the difference between y = mx and y = mx + b). You can 'fake' these affine transformations by augmenting the matrices appropriately. The system below translates points in the x and y directions by Δx and Δy.

$$\begin{bmatrix}x_0+\Delta x & x_1+\Delta x & x_2+\Delta x \\ y_0+\Delta y & y_1+\Delta y & y_2+\Delta y \\ 1 & 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1\end{bmatrix}$$

For consistency, the rotation and translation matrices can also be cast in augmented form:

$$rotation: \begin{bmatrix}\cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1\end{bmatrix}$$

$$scaling: \begin{bmatrix}D & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & 1\end{bmatrix}$$

Note that the associatively rule for matrix multiplication lets us combine these operations into a single matrix that will perform a composite operation. For example, a rotation about the point (1,1) can be represented as a three step process:

1. Translation by (-1, -1): This moves the point that was at (1,1) to (0,0). All other points go along for the ride.
2. Rotation around the origin by an angle θ.
3. Translation by (+1, +1): This moves the point now at the origin back to where it came from: (1,1).

This process can be represented as

$$\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -1\\ 0 & 1 & -1\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_0 & x_1 & x_2\\ y_0 & y_1 & y_2\\ 1 & 1 & 1\end{bmatrix}$$

$$=\left(\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -1\\ 0 & 1 & -1\\ 0 & 0 & 1\end{bmatrix}\right)\begin{bmatrix}x_0 & x_1 & x_2\\ y_0 & y_1 & y_2\\ 1 & 1 & 1\end{bmatrix}$$

The term in parentheses can be computed once, and the process represented by a single matrix. For θ=30° this matrix is…

```
>> theta = 30 * (2 * pi / 360);
>> M1 = [1 0 1; 0 1 1; 0 0 1];
>> M2 = [cos(theta) -sin(theta) 0; sin(theta) cos(theta) 0; 0 0 1];
>> M3 = [1 0 -1; 0 1 -1; 0 0 1];
>> M = M1 * M2 * M3

M =

    0.8660   -0.5000    0.6340
    0.5000    0.8660   -0.3660
         0         0    1.0000
```

