

4452 Mathematical Modeling Lecture 9: Discrete Dynamical Systems

Many processes in the world are not continuous, but discrete. Compound interest, population growth, and feedback control all may be more suitably modeled as a discrete time event. Discrete dynamical systems or *difference equations* leads to *recurrence relations* which must be analyzed. Recurrence relations are easy to set up and solve, using a calculator, a spreadsheet, or a computer. Although you may be able to get a closed form for the solution, you can always just iterate the recurrence relation to determine the behaviour of the system.

Recurrence Relation Basics

A first order recurrence relation has the form:

$$y_{n+1} = f(n, y_n), \quad n = 0, 1, 2, 3, \dots,$$

which is sometimes written as

$$y(x_{n+1}) = f(n, y(x_n)), \quad n = 0, 1, 2, 3, \dots$$

This is first order since the value of y_{n+1} depends explicitly on the previous y_n , but not earlier values like y_{n-1}, y_{n-2} . It is linear if f is a linear function of y_n , otherwise it is nonlinear. It is autonomous if $f(n, y_n) = f(y_n)$. An initial condition $y_0 = \alpha$ is required to initialize the difference equation, and the solution is the sequence of numbers y_0, y_1, y_2, \dots

Aside: A *Difference Equation* would look like:

$$y_{n+1} - y_n = f(n, y_n).$$

Example Let's analyze the recurrence relation

$$y_{n+1} = -\frac{1}{2}y_n + 6. \tag{1}$$

After choosing an initial condition $y_0 = \alpha$, you could just iterate this equation and plot it and try and determine the behaviour that way. However, *Mathematica* has a package which is very similar to DSolve that can solve recurrence relations for you. It is RSolve, and it has to be loaded into the kernel before you can use it. Using RSolve, we find that the solution is

$$y_n = \frac{-4(-1)^n + 2^{2+n} + (-1)^n c}{2^n}$$

This is a family of sequences, where the c can be identified as the analogue of the constant of integration in a differential equation. We can use the initial condition to determine c :

$$y_0 = \alpha = c$$

$$y_n = \frac{-4(-1)^n + 2^{2+n} + (-1)^n \alpha}{2^n}$$

The solution for $\alpha = 2$ is shown in Fig. 1. The sequence of solutions looks like this:

$$\{ \{0, 2\}, \{1, 5\}, \{2, 3.5\}, \{3, 4.25\}, \{4, 3.88\}, \{5, 4.06\}, \{6, 3.97\}, \{7, 4.02\}, \{8, 3.99\}, \{9, 4.00\}, \dots \}$$

We can see an oscillation, and that the solution appears to be approaching 4 as $n \rightarrow \infty$ ($\infty = 10$, in this case...).

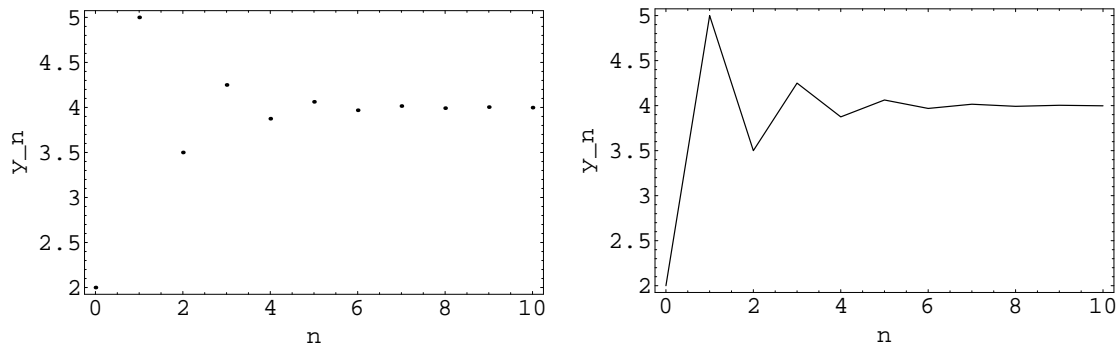


Figure 1: The solution to Eq. (1) when $y_0 = 2$. Typically, any figures you find in the literature will have the points joined for ease of viewing, but really the discrete function is not defined except at the points.

The equilibrium solution is found by solving $y_n = y_{n+1} = f(y_n)$ for y_n . We guessed that the equilibrium solution would be 4 based on our graph, and we can prove that numerically now.

Another way of describing the series is sometimes used for autonomous equations. This method is related to the idea of nullclines we saw in differential equations. Here you let $y_n = x$, plot the function $f(x) = -\frac{1}{2}x + 6$ and the straight line $y_n = x$. The intersection of these two lines will be at the equilibrium point. I will draw this representation for this case in class, but I will include the graph here in Fig. 2.

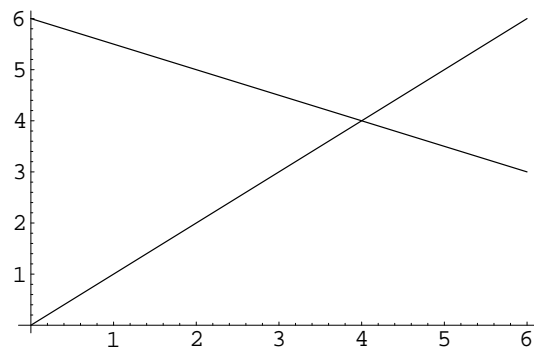


Figure 2: The solution to Eq. (1) when $y_0 = 2$.

Programming Notes

Before we get to chaos, I should say a few words about programming with *Mathematica*. Here is the obvious way to program a recursion relation in *Mathematica*:

```
y[0] := 1/3.
y[n_] := 4 y[n - 1] ( 1 - y[n - 1])
data2 = Table[{n,y[n]}, {n, 0, 40}]
```

This works fine logically, and is alright for small n . However, think about what this is doing. What is happening with this construct is *Mathematica* climbs down the “tree” all the way to the base to get each new y_n ! It is not storing values, so if it wants y_5 it has to recompute y_4, y_3, y_2 , and y_1 (y_0 is simply defined in this case). Then for y_6 it has to recompute all those values! It would be much better (read: faster) if we stored our values as we went, and then *Mathematica* could just look ups each value each time rather than recomputing. Storing each value can be accomplished by the following *Mathematica* commands:

```
y[0] := 1/3.
y[n_] := y[n] = 4 y[n - 1] ( 1 - y[n - 1])
data2 = Table[{n,y[n]}, {n, 0, 40}]
```

Chaos

Virtually any book on chaos begins with simple recursion relations, and chaos becomes readily apparent. Chaos can be thought of as a system which is highly sensitive to its initial conditions. Over short time periods, small changes in the initial conditions do not affect the solution much at all, but the cumulative effect of small differences at each time step results in two solutions which are greatly different over longer time intervals. This obviously has some relation to the sensitivity analysis you have been performing.

Why this chaotic nature is readily apparent in discrete systems is due to the time delay that is built into the system. Figure 3 shows two solutions to the discrete dynamical system

$$y_n = 3.65y_{n-1}(1 - y_{n-1}), \quad y_0 = \alpha, \quad n = 1, 2, 3, \dots \quad (2)$$

for different initial values α . This is a logistic equation which can be used to model population growth. You can see that for short times the solutions remain similar, but for longer times they are different. It turns out that the solution is chaotic because of the choice of 3.65 in our model. It is difficult to determine the behaviour in an analytic sense for this problem, since it is nonlinear. Chaos would tell us that if the growth rate (which is what 3.65 represents) is too large it will be *impossible to make long range predictions on the behaviour of the system* [1].

Comparing Continuous and Discrete Models

What about the continuous analogue of Eq. (2)? You might think that it would be the differential equation

$$\frac{dy}{dt} = 3.65y(1 - y), \quad y_0 = \alpha,$$

but this isn't quite right. Let's take a moment to examine what is meant by the relationship between a continuous and discrete model. It is easiest to begin from the continuous, and work towards the discrete.

$$\begin{aligned}\frac{dy}{dt} &= f(y) \\ \frac{\Delta y}{\Delta t} &= f(y_{n-1}) \\ \frac{y_n - y_{n-1}}{\Delta t} &= f(y_{n-1}) \\ y_n &= f(y_{n-1})\Delta t + y_{n-1}, \quad n = 1, 2, 3, \dots\end{aligned}$$

and we see that our original guess was wrong! We really want to be working in the frame of difference equations, not recurrence relations, when we compare discrete and continuous models.

So if we have the discrete model which is given by a recurrence relation

$$y_n = 3.65y_{n-1}(1 - y_{n-1}), \quad y_0 = \alpha, \quad n = 1, 2, 3, \dots \quad (3)$$

the appropriate continuous model to compare it to is found by the following reduction

$$\begin{aligned}y_n &= 3.65y_{n-1}(1 - y_{n-1}), \quad y_0 = \alpha, \quad n = 1, 2, 3, \dots \\ y_n &= y_{n-1}(2.65 - 3.65y_{n-1}) + y_{n-1}, \quad y_0 = \alpha, \quad n = 1, 2, 3, \dots \\ \frac{y_n - y_{n-1}}{\Delta t} &= y_{n-1}(2.65 - 3.65y_{n-1}), \quad y_0 = \alpha, \quad n = 1, 2, 3, \dots, \quad \Delta t = 1 \\ \frac{dy}{dt} &= y(2.65 - 3.65y), \quad y(0) = y_0 = \alpha\end{aligned} \quad (4)$$

where the discrete model has a time step of $\Delta t = 1$.

The solution to this differential equation initial value problem (Eq. (4)) was found via an interpolating polynomial in *Mathematica*. The solution is graphed in Fig. 3. Notice that it has no chaotic, oscillatory behaviour as that observed in its discrete analogue.

Let's compare the discrete and continuous cases in a region where chaos is not present for the discrete case. This time I am going to change the discrete case to make it match up with the continuous logistic model.

$$\text{Discrete: } y_n = 2y_{n-1}(1 - y_{n-1}) + y_{n-1}, \quad y_0 = 1/10, \quad n = 1, 2, 3, \dots \quad (5)$$

$$\text{Continuous: } \frac{dy}{dt} = 2y(1 - y), \quad y_0 = 1/10, \quad (6)$$

The two solutions are compared in Fig. 4. Notice that now that the discrete model is not in a chaotic regime, the behaviour is very similar to that of its continuous analogue. However, the quantitative behaviour is different.

The quantitative difference we see in Fig. 4 is expected, since the two models should be similar only in the limit as $\Delta t \rightarrow 0$. The difference between the two models can be explained by the effects of the time delay.

We also can see how to decrease the time delay, and bring the models into closer agreement. This is essentially a rescaling of time, and both the discrete and continuous cases have to be changed. Equations (7) and (8) are compared in Fig. 4.

$$\begin{aligned}y_n &= 2y_{n-1}(1 - y_{n-1})\Delta t + y_{n-1} \\y_n &= 2y_{n-1}(1 - y_{n-1})0.1 + y_{n-1} \\y_n &= 0.2y_{n-1}(1 - y_{n-1}) + y_{n-1}, \quad y_0 = 1/10 & (7) \\ \frac{dy}{dt} &= 0.2y(1 - y), \quad y(0) = 1/10 & (8)\end{aligned}$$

Finite Difference Calculus

Just as there is a differential calculus for continuous functions, there is a difference calculus for discrete functions. It is useful to know when you have a set of data points, since then you simply *must* work with a discrete model (unless you choose to fit a continuous function to the data, of course, but that presents its own challenges).

You have *forward differences* and *backward differences*, and can define derivatives and integrals over a discrete set of points. Typically, the time step is assumed to be uniform, however, you can generalize to a nonuniform time step if you need to.

This would be a good topic for a final project, and I could give you a real world data set to work with (well, slightly real world).

References

- [1] R.M. May, *Biological Populations with Nonoverlapping Generations: Stable Points, Stable Cycles, and Chaos Science* **186** pp 645–647 (1974).

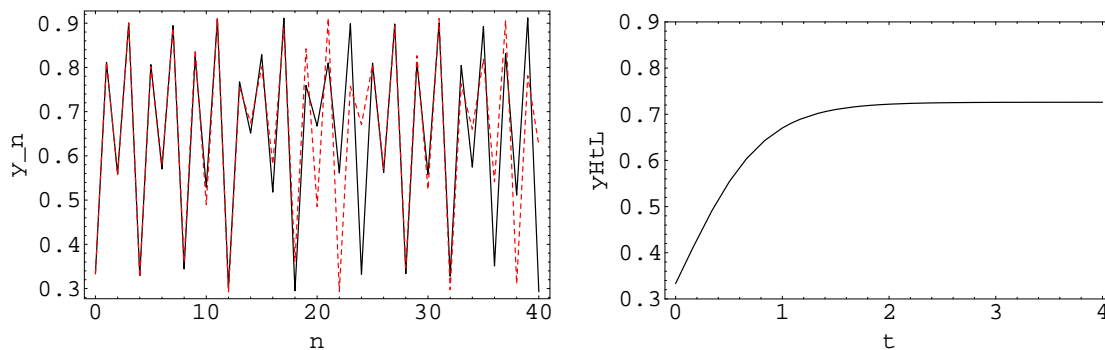


Figure 3: On the right: The solution to Eq. (2) when $y_0 = 1/3$ (solid line) and $y_0 = 1/3 + 0.001$ (dashed line). This shows evidence of chaotic behaviour. On the left: The solution to Eq. (4) when $y_0 = 1/3$. Changing the initial condition by a small amount makes an insignificant change in the solution, so chaos is not apparent in the continuous model.

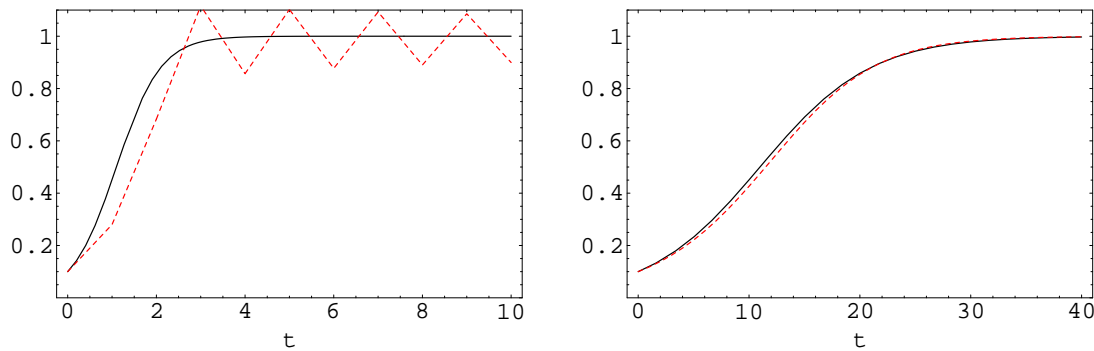


Figure 4: On the left: The solution to Eq. (5) (dash) and Eq. (6) (solid). The time delay in the discrete model causes the solution to overshoot the continuous equilibrium, and then settle into an oscillation about that continuous equilibrium. On the right: The solution to Eq. (7) (dash) and Eq. (8) (solid). The time delay in the discrete model has been reduced, increasing the agreement between the discrete and continuous cases.