John Balwit

The Logistic Equation, the Logistic Map and Chaos

In this series we will explore several ideas that lie at the heart of complexity science.

- Iteration as a source of complexity.
- The use of mathematical models to describe real world events and relationships.
- Continuous vs. discrete time.
- Sensitivity to initial conditions.
- Determinism and fate.
- The logistic map as a path to chaos.
- Feigenbaum's constant

The models that we will use to explore and illustrate these ideas include:

Logistic equation-based model
Logistic emergent model
Iteration 101
Iteration explorer
Toggle
Mandelbrot Explained
Logistic discrete time model (bunnies)
Web Diagram
Bifurcation Diagram (with zoom)
Sensitivity to Initial Conditions
Universal Patterns: Feigenbaum's Constants

Repetition on on the road to complexity

Very early in human history people recognized that repetition is a deep and ubiquitous feature of life. The cycles of the days, the season, the rise and fall of individual fortunes and the fortunes of whole peoples, the succession of generations. All of these things share the common feature of repetition and variation within that repetition. In this section we will explore some of the ways that repetition or, *iteration*, as it is more formally called, sometimes gives rise to surprising forms of complexity.

It is quite obvious that the repetition of a process will result in quantitative change--that is, an accumulation or loss of some quantity or characteristic. On the other hand, it is surprising when a repetition of a simple process results in a qualitatively new situation. We will see later in this section that this is not only possible but an inherent aspect of some of the fundamental processes of living system and other interesting systems.

Iteration (notice that I am switching to repetition's formal name) usually involves repeating some process over and over again with small regular changes at each step. Counting aloud with the natural numbers (1,2, 3, ... and so on) serves as a familiar example of iteration. We effortlessly add 1 to the number that we just said and say a new number. Then we add one to that number, say it and repeat the process as long as we please.

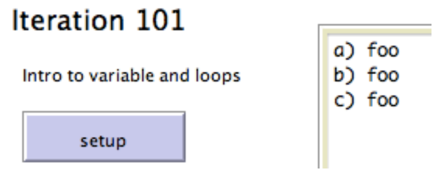
Other examples of iteration might include the process that we use to find a precise decimal equivalent for 22 / 7. If we do this by hand, we follow a procedure that is mix of division, multiplication, and subtraction. Frequently iteration of this sort is tedious and error-prone. Human attention spans are short and vulnerable to distraction. Fortunately, there is another way. Iteration is a area in which computers excel.

We begin our thinking of iteration with a few very simple models that demonstrate how computer (specifically, how NetLogo) handles iteration.

Iteration 101

The following models are simple and useful to students new to NetLogo. Others should feel free to skip this section and go directly to the section on the Logistic Equation.

The first demonstration simply introduces users to several ways that iteration is accomplished in NetLogo. For beginning programmers, one of the first and most fundamental concept is the notion of a variable. This models illustrates how a single variable can hold various values as a program runs.



```
a) foo
b) foo
c) foo
```

This model illustrates use of the “repeat” primitive which simply repeat an action a given number of time . It also introduces the “output-window”, an interface widget that displays information that is sent to it.

The use of special variables that change during iteration, *external iterators*, is explored in this model as well. These *iterators* are useful in helping programmers travel through or *traverse* a set of items.

Finally, NetLogo lists are introduced. Lists are a handy way of holding groups of related items that you may want to deal with systematically. A list might include a group of numbers that you might want to sum, the names of all the students in a class, or even, a list of lists--a list contain the names of friends, their addresses, and their birthdays.

NetLogo relies heavily on *implicit iteration*. Each time you ask turtles or patches to do something, you are asking a group or in NetLogo-ese, an *agentset*, to do something. NetLogo selects an agent at random from the pool of agents in the agentset and asks the agent to do the action described. When all agents have had heir turn, the *ask* block is finished.

NetLogo *“foreach”* loops also use implicit iteration. In this case a special primitive “?” is used to represent the current item that is “on stage”. Unlike ask statements, foreach loops proceed in an orderly, sequential fashion from the first to the last item in the list.

```
to loop-over-a-list
  clear-output
  foreach [ "a" "b" "c"] [output-print ? ]
end
```

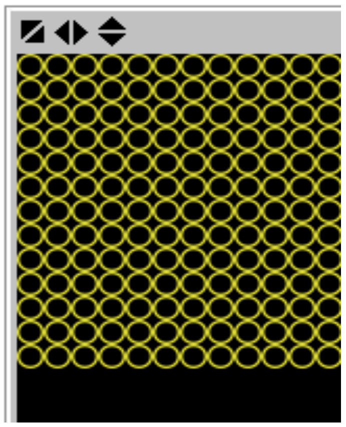
Toggle

The second model, Toggle, illustrates how interesting and surprising behavior can sometimes result from a simple action repeated over and over. In Toggle, we have a field of light bulbs numbered 0 through 2400. Each light bulb knows its own number. As we step through the numbers 0 through 2400, we ask each light bulb to toggle its ON/OFF state if it is perfectly divisible by that number*. Lights that are ON turn OFF and lights that are ON turn OFF if divisible by the current number. The result is a bunch of flashing lights. Patterns march across the fields and when the dust settles some lights are ON and others are OFF. Can you guess which ones are ON? Hint: It has nothing to do with prime numbers.

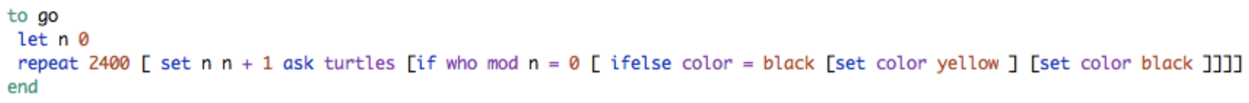
Toggle

Iteration in Action

A field of numbered light bulbs. Turn them OFF their number is divisible by 1, turn them ON if their number is divisible by 2, turn them ON if divisible by 3, turn OFF if divisible by 4, and so on....



Toggle illustrate several things. First, simply applying the same rules with very slight changes--here we are just counting up--can creating interesting and potential useful behavior. Who would have guessed that you could find all the perfect squares simply by flicking light bulbs on and off? (Although, I admit it seems like a cheap trick after the fact!) The second observation is that iteration gives us a lot of bang for the buck. The entire Toggle program (aside from some introductory eye candy, is really only a couple of lines.



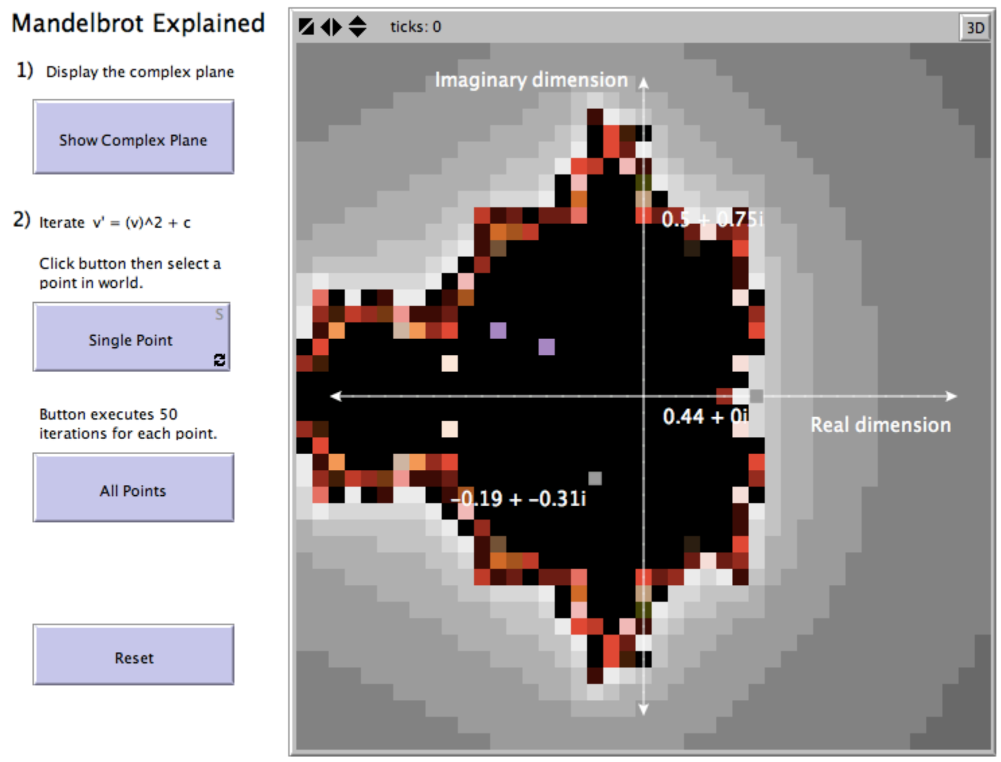
```
to go
 let n 0
 repeat 2400 [ set n n + 1 ask turtles [if who mod n = 0 [ ifelse color = black [set color yellow ] [set color black ]]]]
end
```

That's it. The whole thing! Iteration is the very heart of computer programming.

Finally, Toggle illustrates how patterns and answers often emerge from simple processes repeated over and over. More on this latter.

Mandelbrot Explained

Mandelbrot Explained is a model that focuses on how iteration produces the spectacular and iconic images found in fractals. Complexity Explorer has an entire section devoted to fractals for students interested in pursuing this topic in depth. The purpose of this model is simply to demonstrate the iteration that lies at the foundation of these images.


Mandelbrot Explained

1) Display the complex plane

Show Complex Plane

2) Iterate v' = (v)^2 + c

Click button then select a point in world.

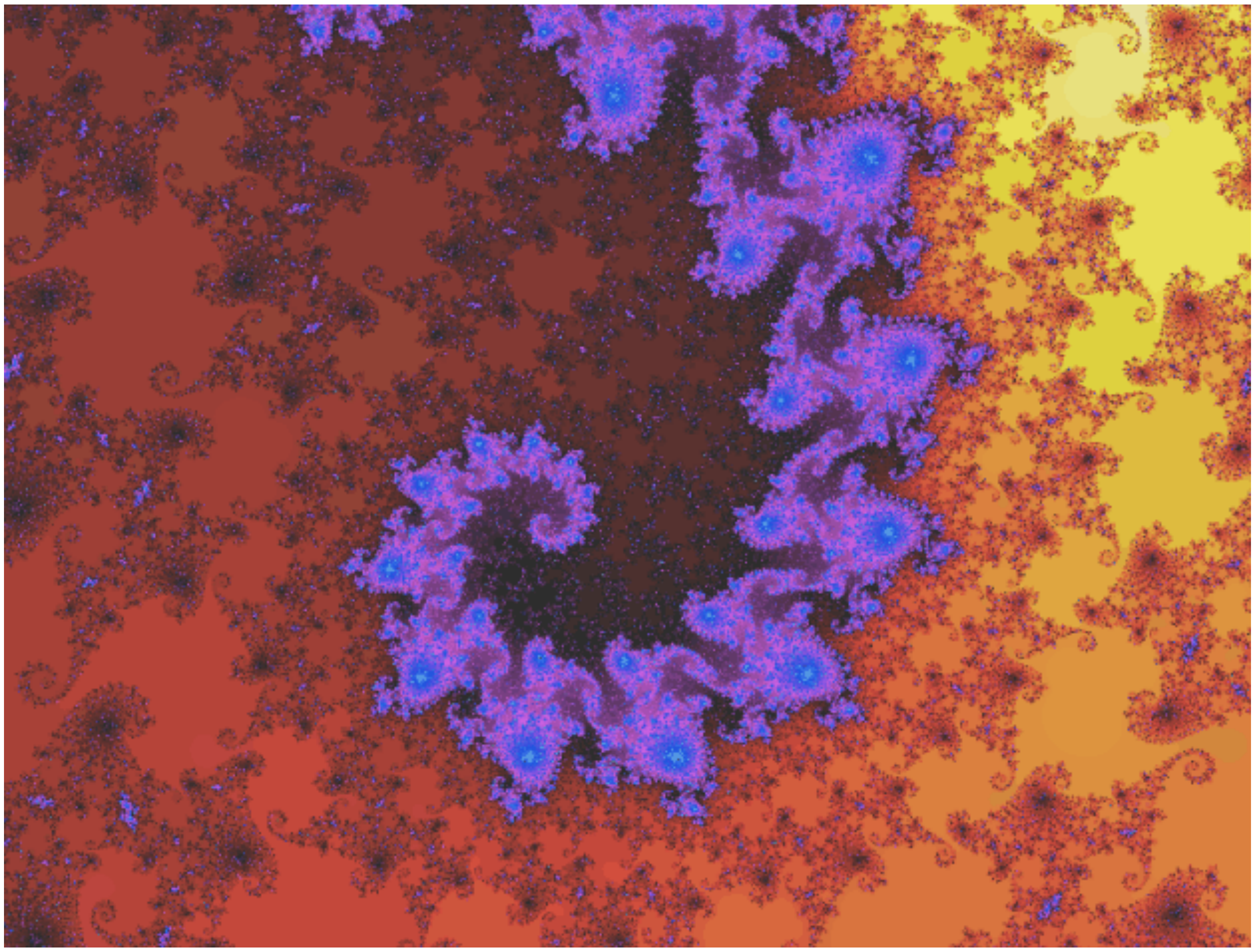
Single Point

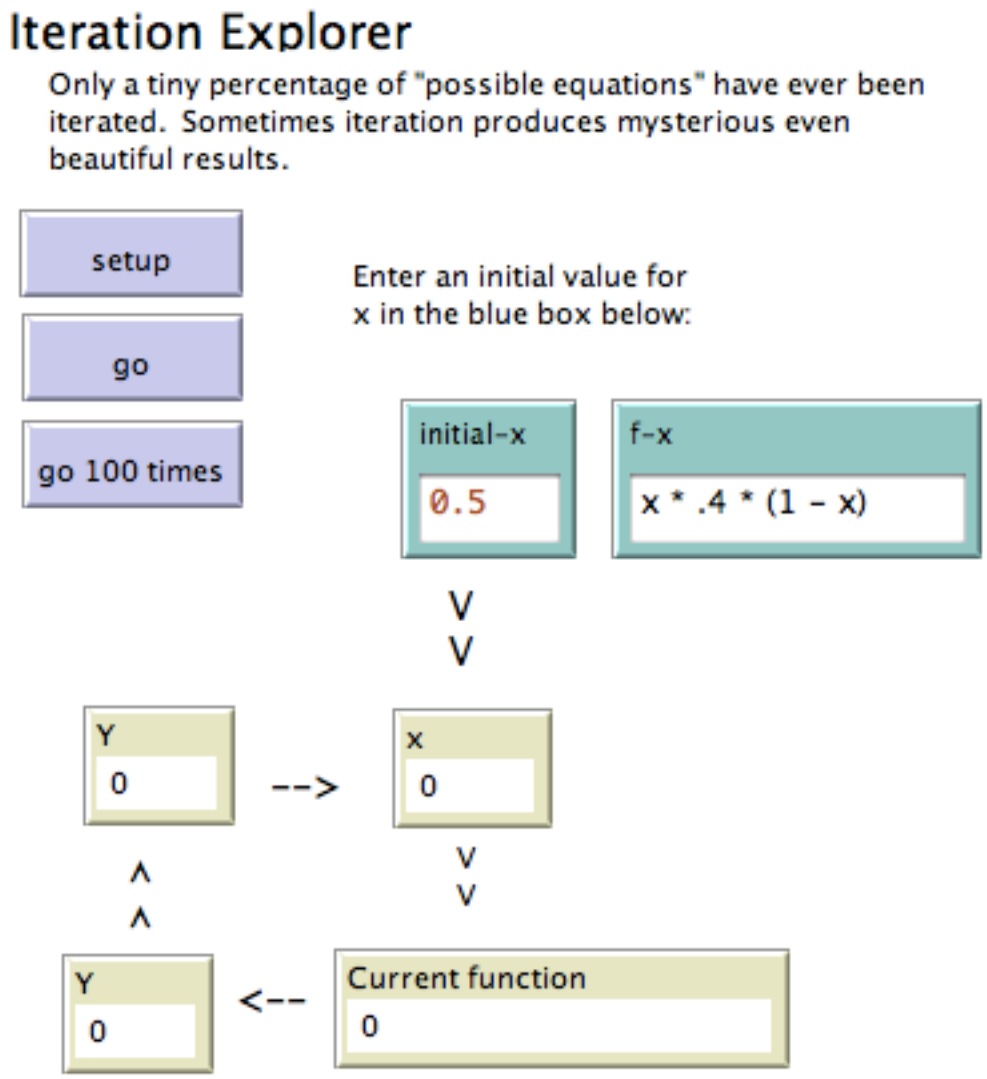
Button executes 50 iterations for each point.

All Points

Reset

Mandelbrot Explained illustrates the complex plane and show iteration can be used to sort cells/values into those that, when run through an iterative process, exceed a set threshold value from those that don’t. Cells that lie on the edge of these two sets define create the beautiful, complex images that are the familiar province of fractals.


Example of high resolution fractal created with FractInt freeware.


Iteration Explorer

Only a tiny percentage of "possible equations" have ever been iterated. Sometimes iteration produces mysterious even beautiful results.

Enter an initial value for x in the blue box below:

initial-x: 0.5

f-x: x * .4 * (1 - x)

Y: 0

x: 0

Current function: 0

Y: 0

Iteration Explorer is final demonstration in the iteration sub-sequence. This tool allows users to plot the consequences of iterating an equation. Iterating an equation means starting with an initial value, running the value through a function, noting the result and then *using* the result as the next value run to be through the function. An example should make this clear:

Given that our initial value is 1 and our function is x + 2.
Step 1: 1+ 2 -> 3
Step 2: record 3
Step 3: use 3 as new input, 3 + 2 -> 5
Step 4: record 5

Step 5: use 5 as new input, 5 + 2 -> 7 and so on.

Obviously, this function will just continue to growth by a steady rate. Our answer includes points: the numbers 3, 5, 7 etc. and not the points in between these numbers. Functions that produce discrete results like this--where the results of the function is a set or sequence of numbers are often called *recurrence equations* or *difference equations*.

When iterating equations, it is easy to find equations with quickly grow to an unmanageably large size. It is also easy to find equations that dwindle away to nothing after a few iterations. The most interesting equations are the class of equations that meander around for a while, sometimes a great while, before settling down to a particular value. These are the sort of equations that we will want to hunt for with the iteration explorer.

Although explorer can be adapted to iterate any function, we will considering a particular equation, the logistic equation, and the iteration of its iterated or discrete form. This difference equation is represented mathematically as:

$$x_{n+1} = \mathrm{r}$$

The x_{n+1} term represents the new value and also the value that will be used a x_n in the next iteration.

The result of iterating the logistic difference equation is curious insofar as the result depends upon the value of r. For small values of r, the iterated result goes to zero. For slightly higher values, the result goes to a constant number, at even higher values of r the result oscillates between two point. With even higher values of r the oscillations are between 4, 8, 16 etc. unique values.

The behavior observed in the explorer is also evident in natural systems that can be modeled by the logistic equation in discrete time. Fish population with very low growth rates (lower than the replacement rate) go quickly extinct no matter what the initial population. With a growth rate slightly above the replacement rate, populations behave as we might expect--higher growth rates yield higher populations. At a certain point, the growth rate is so high that over-population in one generation results in a smaller population the very next generation. This smaller generation grows rapidly in the absence of competition and a boom/bust cycle commences. The boom bust cycles have the potential to become quite complicated. It seems plausible that much of the intricacy and pattern in the natural world is the result of chaotic dynamics like that observed in the logistic map.

Iteration 101 exercises:

1. Starting from a blank NetLogo screen, create a program that loops 100 times then quits.
2. Create a NetLogo program that loops over the list ["Tom" "Dick" "and" "Harriet"] using a foreach loop and the special "?" character. The program should create and name (label) the turtles using the list. Consider improving the program so that we don't have turtles named with the silly name "and".

   BTW: The other three names are NOT silly. They are the names of the three tortoises that Darwin bought back from the Galapagos (orig. "Tom Dick and Harry" but Harry was discovered to be of the gentle sex and so renamed "Harriet". Harriet survived Darwin by 124 years, dying in Australia in 2006.

3. Use the awkward but useful "n-values 10 [?]" primitive and the confusing but powerful "map" primitive to do something useful and powerful: A numbered list of the first 10 rows of Pascal's triangle.

Iteration is the bread and butter of computer programming. It also has a prominent role in certain kinds of mathematics. Calculus, with its emphasis, on limits, can be presented as a way of thinking about the sums or slopes that are changing under an iterative process. Iteration is "automated" in mathematics in differential equations which are used to describe changes in rates when the change in a quantity depends (in part) on the quantity in question. Because is typically the case that change in real-world quantities *is* closely tied to the levels of those quantities at a previous point in time, differential equations are one of the most frequently used tools in the physical scientist's toolbox.

A close cousin of the *differential equation* is the *difference equation*. Whereas the differential equation (when used as a model of real world phenomena) makes the assumption time flows continuously and is infinitely divisible, the difference equation makes the assumption that time happens in discrete lumps: hours, days, years, generations.

There are wonderful visual tools that students can use to explore the world of modeling natural phenomena with differential equations. System dynamics (SD) "stock and flow" models uses "stock" (quantities) and "flows" (rates of change) to create models of dynamic situations: changing demographics, the spread of disease, adoption of technology etc. NetLogo has a SD stock and flow module that is included with the standard distribution. Although we will not focus much attention here on the system dynamics, it may be interesting for students to familiarize themselves with these methods. The realization that complex phenomena with reinforcing and inhibiting feedback can be captured by a set of equations is quite astonishing the first time one encounters it.

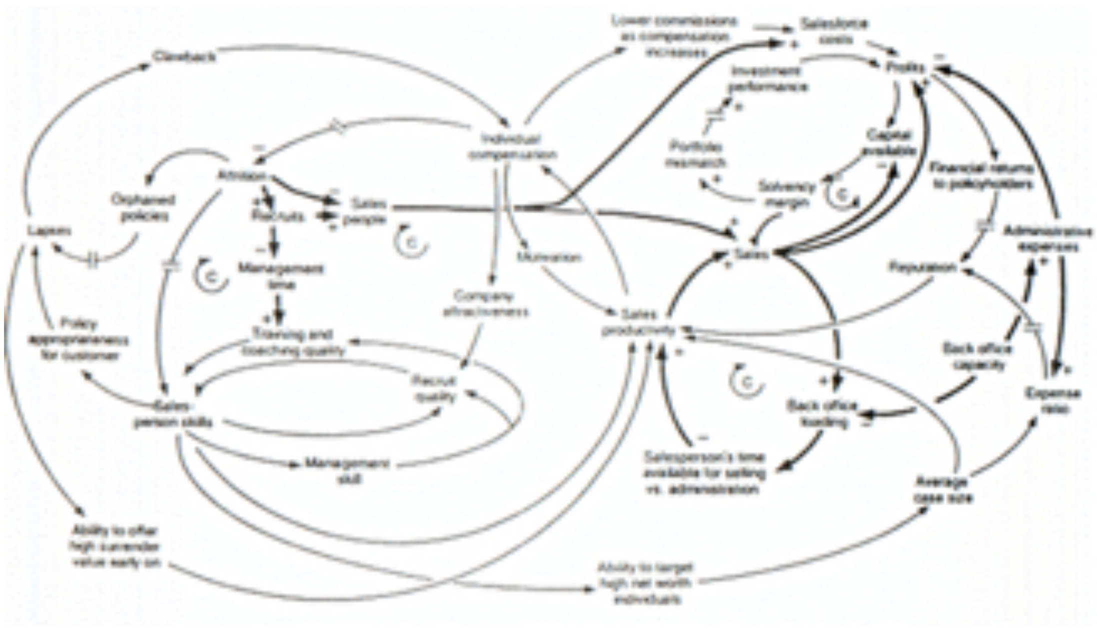


Fig. 6 Typical systems dynamic model showing positive and negative feedback.

Solitary equations in continuous time have simple behavior. Over time these functions grow, shrink, or approach an asymptote (a certain value that the function never quite achieves).Sets of interdependent equations can generate more complex behaviors: Oscillations, limit cycles, spiraling growth etc.

When we move to difference equations in discrete time, truly complex behavior is available with only a single equation. Whereas the differential equations produce a continuous output of values, difference equations (aka recurrence relations) produces a sequence of numbers as output.

The demonstration model, Iteration Explorer. helps students observe the consequences of iteration a equations in discrete time.

Iteration Explorer Exercises

1. Find two equations other than the examples provided that have “interesting behavior” ie. don’t simple grow infinitely, shrink or approach a value.

2. What are the minimum requirement for “interesting behavior”.

3. Starting with a blank NetLogo screen create a program that prints a sequence to an output window.

   Modify the above program so that it uses the built in coordinate system in the world to plot values.

The Logistic Equation

The logistic equation is a mathematical model of exponential growth in the context of a constraint to growth: limited food or limited space. The constraint imposes a limit to growth and establishes a “carrying capacity”. The logistic equation was first published by Pierre Verhulst in 1845 and has found wide application in artificial neural networks, biology, biomathematics, demography, economics, chemistry, mathematical psychology, probability, sociology, political science, and statistics. Technically, the logistic function is the solution of the simple first-order non-linear differential equation.

If we consider the logistic equation in discrete time--for example, a situation where next year’s population depends on this year’s ending population--we arrive at something called the *logistic map*. The logistic map offers what has become the canonical example of a system that follows simple rules yet arrives as behaviors that are surprisingly complex. Often regarded as a “path to chaos”, the logistic map illustrates the dynamics of a deterministic system generating pseudorandom behavior. The logistic map has intrigued people since the late 1940’s and the equation continues to be of great interest and pedagogical value as an illustration of various principles that lie at the heart of complexity science.

John Balwit

Introduction to Concepts

During the 19th century, scientists were wrestling, perhaps for the first time in human history, with the somber observation that we live in a world of limited resources. Populations were growing and the land fit for agriculture (feeding people) was limited. Economist Thomas Malthus first expressed these ideas in a work called *An Essay on the Principle of Population.*

> "The power of population is so superior to the power of the earth to produce subsistence for man, that premature death must in some shape or other visit the human race. The vices of mankind are active and able ministers of depopulation. They are the precursors in the great army of destruction, and often finish the dreadful work themselves. But should they fail in this war of extermination, sickly seasons, epidemics, pestilence, and plague advance in terrific array, and sweep off their thousands and tens of thousands. Should success be still incomplete, gigantic inevitable famine stalks in the rear, and with one mighty blow levels the population with the food of the world".
>
> —Malthus T.R. 1798. *An essay on the principle of population*. Chapter VII, p61[3]

These ideas had profound effect on the thinkers of the day. Darwin and Wallace independently arrived at their theories of natural selection after reading Malthus. A less well-known mathematician, Pierre Verhulst, developed an equation that modeled the interactions between population growth and the forces that limit growth. This "Logistic Equation" as it is now known, can be stated in words quite simply:

"The future size of the population (change in population) is affected positively by the growth rate (how many offspring are born) and negatively by consequences of overcrowding or resource depletion."

We can say this compactly mathematically (Fig.1):

Fig. 1

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$$

Bear with me as we step through this equation. It is not as complicated as it might appear to those unfamiliar with mathematics. Verhulst's equation shows how populations change over time--that's the **dP/dt** part, its just a number -- given different intrinsic growth rates, **r** (called the Malthusian parameter) and different initial populations, **P** , when limited by some constraint to growth or carrying capacity, K. The equation is called a first order non-linear ordinary differential equation.

Even if you are unfamiliar with differential equations it is relatively easy to gain an intuition about how this works:

Consider the extreme situations. When the population, P, is small, the second term, is essentially equal to 1. Multiplying by one doesn't change things. Consequently the change in population is simply equal to the existing population times the growth rate, r. This results in rapid growth for any values of r greater than one since the new larger population is immediately used as the existing population and again multiplied by the growth rate.

As the population grows the second term starts to matter. As population gets closer to the carrying capacity, K, the second term, (1 - P/K), gets closer to zero. Multiplying by zero or very small numbers results in zero or very small numbers. This has the effect of essentially shutting down growth.

.

The second term (as it approaches zero) corresponds to slowed growth due to overcrowding or “reaching the carrying capacity”.

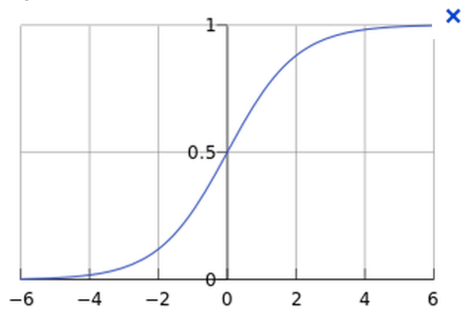
Note: The logistic function is sometimes simplified to:

$$\frac{dx}{dt} = rx(1-x),$$

X is not the population directly. In this case x represents a slightly more abstract concept, namely, “percentage of max. population” or the population divided by the carrying capacity. This is the form of the equation that we will be using in the models below.

The situation can be summarized with an idealized graph (Fig. 2) that represents the situation of early rapid growth followed by a leveling off. This is called, understandably, the S curve or, if you prefer your alphabet in Greek, the sigmoidal curve.

Fig. 2

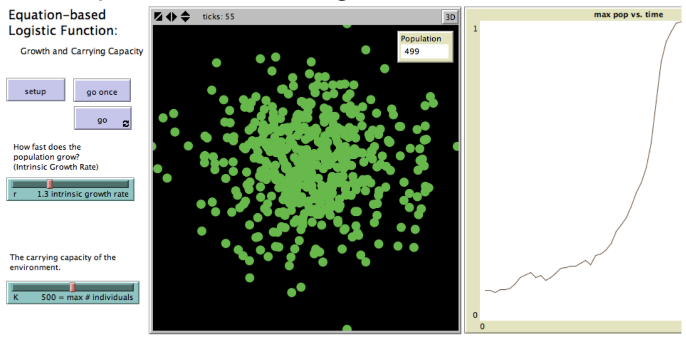


I have lingered over the description of the mathematics here because the logistic equation can serve as a bridge to understanding how mathematical models can be used to represent biological (or other) systems.

Modeling in NetLogo

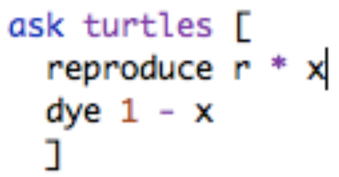
Two NetLogo Models are introduced which illustrate modeling a biological system with a carrying capacity.

The Equation-based Logistic Function


Equation-based Logistic Function:
Growth and Carrying Capacity
setup
go once
go
How fast does the population grow? (Intrinsic Growth Rate)
r 1.3 intrinsic growth rate
The carrying capacity of the environment.
K 500 = max # individuals
ticks: 55
3D
Population 499
max pop vs. time
1
0
0

The first model uses the Logistic equation explicitly and demonstrates how to embed equations directly into NetLogo. This is easy to do in NetLogo and in many cases quite useful. This model is called Equation-based Logistic Function.

Creating NetLogo models of mathematical models is surprising easy. In this case, the two lines of code



```
ask turtles [
  reproduce r * x
  dye 1 - x
  ]
```

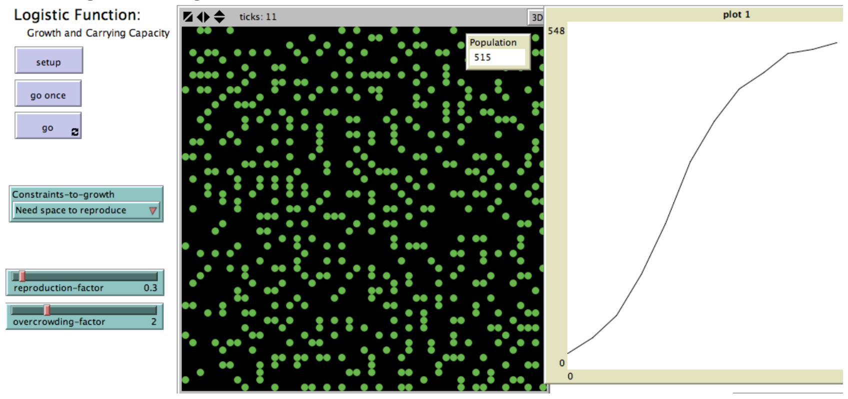
are responsible for most of the behavior of the model. X, however, is determined by a *global* count of the agents (divided by maximum population possible)   In actual environments, individuals rarely have access to information about the global situation and , consequently, behavior is based on local interactions. Because of this, a model that relies only on local information would be give us more confidence that we are seeing the “right behavior for the right reason”. This idea is called the *structural validity* of a model. NetLogo makes it relatively easy to create models which highlight the interactions of lower-level relationships which give rise to the larger scale phenomena. The large scale behavior is sometimes called *emergent behavior*. The nature and study of emergent behavior is one of the primary motivations for research in complexity science.

A second model reproduces the behavior of a logistic function from local constraints in the growth and death rates. This model is called the Emergent Logistic Function.

This model illustrates what happens when reproduction depends upon available space (or available food). The familiar S-shaped curve readily emerges from these scenarios. Decomposing the situation into constraints that have parameters themselves (how much space to reproduce, how much food needed to reproduce) supports fine-grained investigations of real world situations.

John Balwit

Emergent Logistic Function



As it turns out, the logistic function has broad application in various fields. In biology, as we have seen, it models growth under constraint. The logistic function is also used in biomathematics, economics, chemistry, psychology, probability, sociology, political science, and statistics. It can also be used to model tumor growth in cancer patients. In economics it can be used to understand how innovation spreads through a society. The logistic function also finds application in a branch of machine learning called neural networks. In this domain it is useful because it represents a function that can classify inputs and provide some feedback about its confidence of the classification (via an easy-to-calculate derivative).

We have examined the logistic equation as it was first presented to the world by Verhulst (and later by Lotka and others). Equations that, like the logistic equation, connect values and constants to differentials (rates of change) of those same values are called *differential equations*. To work with them we typically use calculus and the idea that we can divide time (and the number line) up into infinitely small pieces. This idea is called *continuous time*. Continuous time and continuity generally is typically assumed in the mathematics of functions that most of us encounter early in our math educations.

Exercises for with the equation based and emergent models:

1. Modify the equation-based model so that it would reflect the consequences of “inbound migration” of a certain number of animals each season. Each season a certain number of animals are added to the population from and external source.
2. Modify the emergent model to include other constraints (food or suitable mates). Can you create a situation where the animals need space to reproduce but too much space decreases the likelihood of reproduction? This situation is quite important in real ecological systems and is called frequency dependence.

Discrete Time

The assumption of continuous time is not always warranted in real-world situations. We know that much of the phenomena that we observe in nature falls into discrete “bins”. Mothers, for example, have a definite number of offspring: 1, 2, or 3 but never 2.8 children. Populations typically change periodically as a result of single annual “birthing season”. We are taught to number our days, seasons, and generations and our numbering uses the “natural” or counting numbers rather than the real numbers (decimal) numbers.

The assumption that time and events happens in definite “chunks” is called *discrete time*. When we work with a version of the logistic equation in discrete time we start to see some very surprising behavior. This behavior and what it can teach about complexity is the primary reason that we are interested in the logistic equation here.

Here’s the discrete time version of the logistic equation (called a quadratic recurrence equation or the logistic map):

$$x_{n+1} = r\,x_n\,(1 - x_n),$$

When we work with the discrete time version of the logistic equation is called a difference equation. Using this equation requires that you are familiar with the concept called iteration. Iteration simply means doing something over and over again (usually with some small change introduced at each loop). Iteration is one of the areas in which computers excel. The NetLogo Iteration models offers an opportunity to explore iteration generally and also allow us to look specifically what happens when we iterate the logistic map in discrete time.

To iterate the above equation, we start with a value for x between 0 and 1 (0.5 is a good place to start) and a value of r between 0 and 4 (2.0 is a good value to start with) and solve the right side of the equation. The result goes into the left side and becomes the new x for the next iteration. Obviously this is a tedious affair and it is no wonder that no significant mathematical progress was made in this area prior to widespread availability of computers.

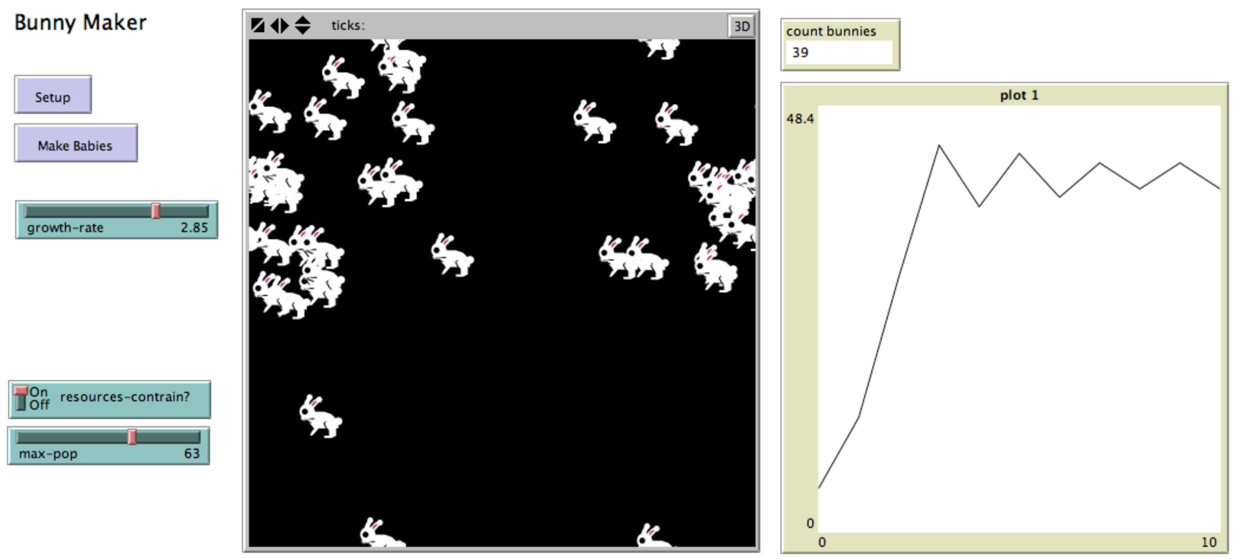
Our interest here is what happens to x after many (as many as 10,000 or even more!) iterations. Does x converge? Does is oscillate? If so, with what period?

Bunnies

The Bunnies model is a straightforward visualization of what happens as we iterate the logistic map. The model illustrates both unfettered exponential growth and growth under constraint.

Logistic Function/ Logistic Map Write Up DRAFT

John Balwit



Exercises for Bunnies

1. Explore and record values of the growth rate for particular values in the following ranges:
   a. 0 - 1.0
   b. 1.0 - 2.0
   c. 2.000 - 3.000
   d. 3.00000 - 4.00000 (note the increased precision!)
2. Can you find a value for r that has a period of three?
3. I forgot to label the axis on the graph. Edit the graph (click on it) and label the axis with “Time” and
4. “Population”
5. Advanced: Add code so that the model doesn’t plot values until the values settle into a pattern (remove initial transient values).

Visualizing the Logistic Map

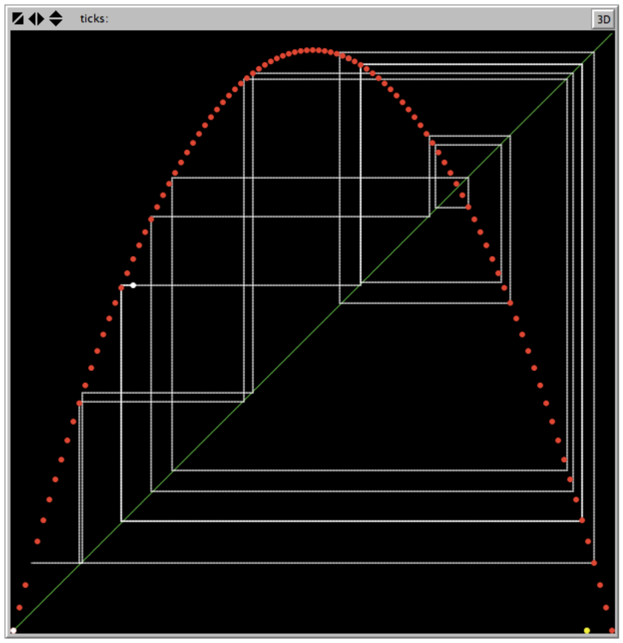
John Balwit

The results of the increasing r and iterating the difference equation, $x_{n+1} = rx_n(1 - x_n)$, can be visualized in several ways. Several of the models presented here allow students to understand graphically the process of iterating the equation and visualize the converge of the equation on various values.

The Web Diagram

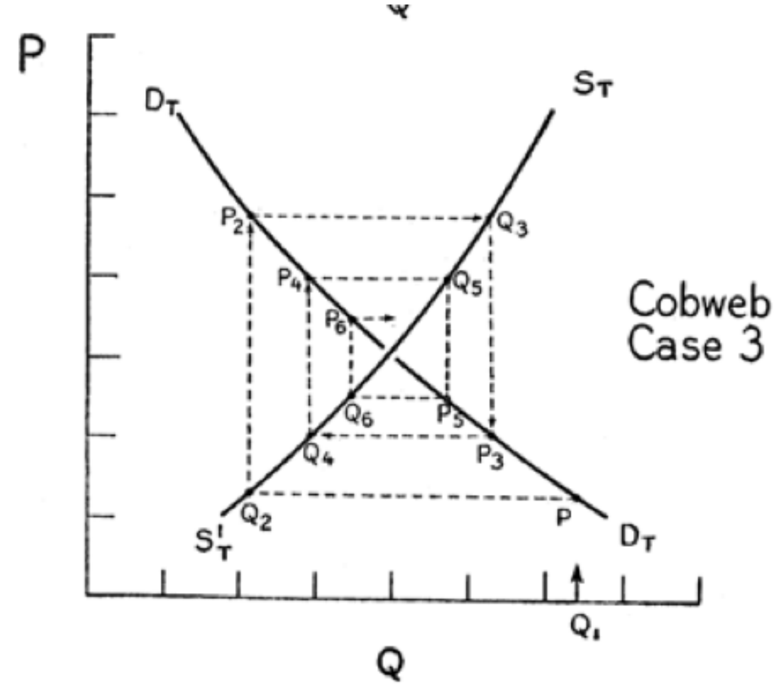
The web diagram has become an icon for complexity science. It provides a simple means of visualizing the patterns that emerge as a consequence of iterating the logistic difference equation. The technique involves:

0) Label the axes: $x_{\text{(x axis)}}$ -> 0-1 and $x_{i+1\text{(y axis)}}$ -> 0-1
1) Draw the parabola that represents all possible solutions xr(1 - x) **for a given r** in the range of 0 - 4.
   Note: Simple algebra applied to xr(1 - x) yields $-rx^2 + rx$, a parabola.
   Remember. We get a different parabola for each value of r.
2) Draw the identity line (y=x), a diagonal passing through the origin
3) Start on the x axis at your initial x value and draw a **vertical** line until you reach the parabola. *This is the result of the first iteration (generation) AND the result that we will use as our input in the next generation.*
4) Now draw a **horizontal** line toward (and until you hit) the diagonal.
5) Now draw a **vertical** line toward (and until you hit) the parabola.
   Step 4 and 5 essential converts the output into the input--think about it.
6) Repeat steps 4 and 5 above until a pattern emerges.



This is a satisfying exercise to perform with a ruler and graph paper. The paper version has the advantage of demystifying the process. it is recommended that students encounter the web diagram initially in the paper and pencil version.

Sidebar: The history of cobweb diagram precedes the discover of the popularization of the logistic map. However, it grows out of a recognition of chaotic dynamics in the field of economics. Mordecai Ezekiel, an early 20th century economist, advanced his “cobweb theorem” to explain the selfperpetuating fluctuations of prices in agricultural markets.



"...Once allowance is made for the fact that in the real world functional adjustments take time and different forces in the system may operate with different 'velocities of adjustment' it may become possible to construct cases--under the assumption that ruling prices are always expected to remain in operation...where the successive reactions lead away from, rather than approach, an equilibrium position."

The assumptions underlying cobweb phenomena are lags in responses and "static expectations." (Kalder, 1934)

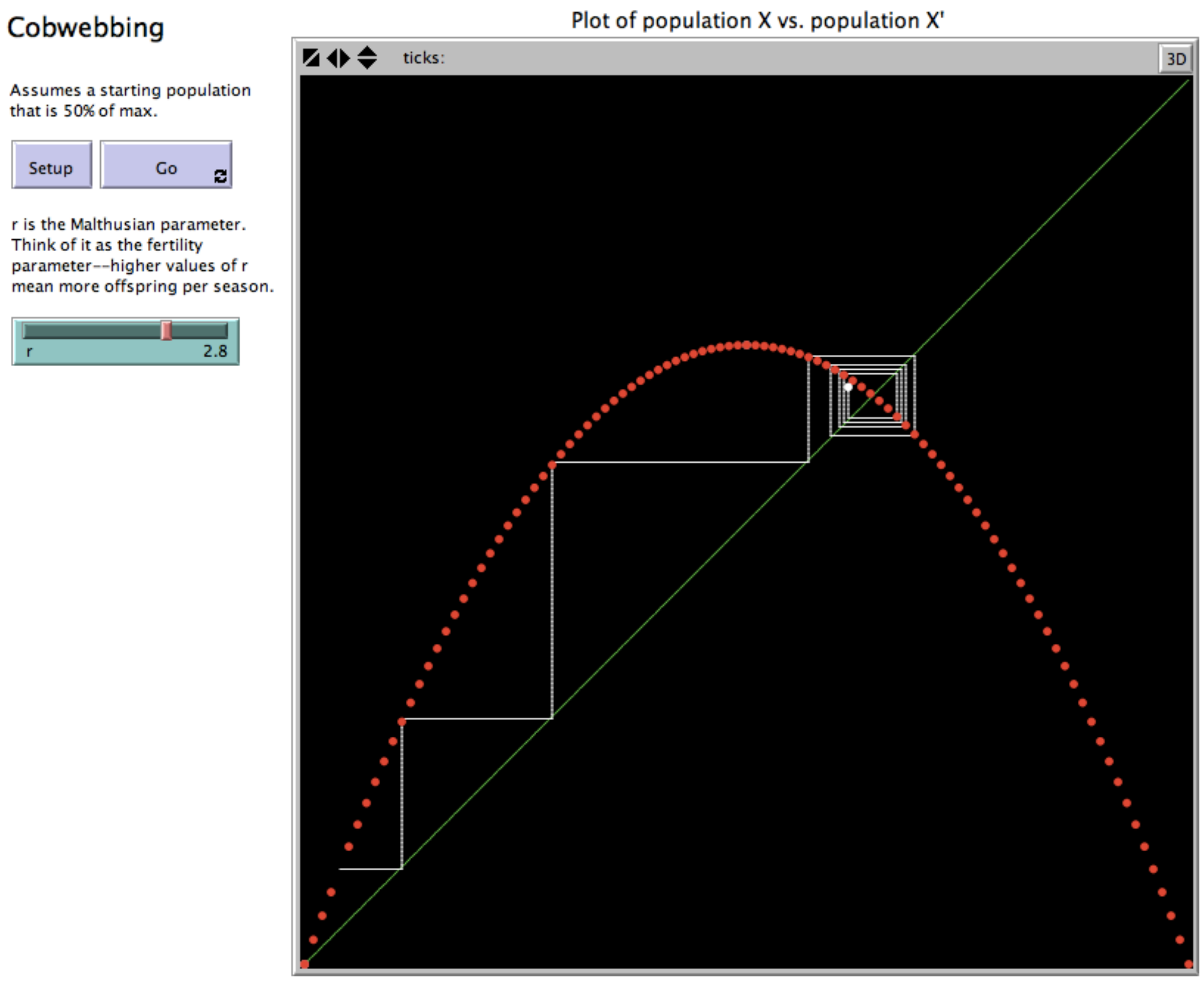
Exercise:

1. Get a piece of graph paper and a ruler.
   a. Make a graph and label horizontal axis x and vertical axis x'
   b. Sketch a diagonal (the identity line)

c. Create a symmetrical parabola with a maximum that doesn't go above the diagonal (it doesn't matter if it is actual a particular parabola just that it is symmetrical)
d. Follow step above with a starting value of x = .25
e. Repeat the above steps for two more parabolas: one that just crosses the diagonal and one parabola with a maximum near x' = .9

--end sidebar--

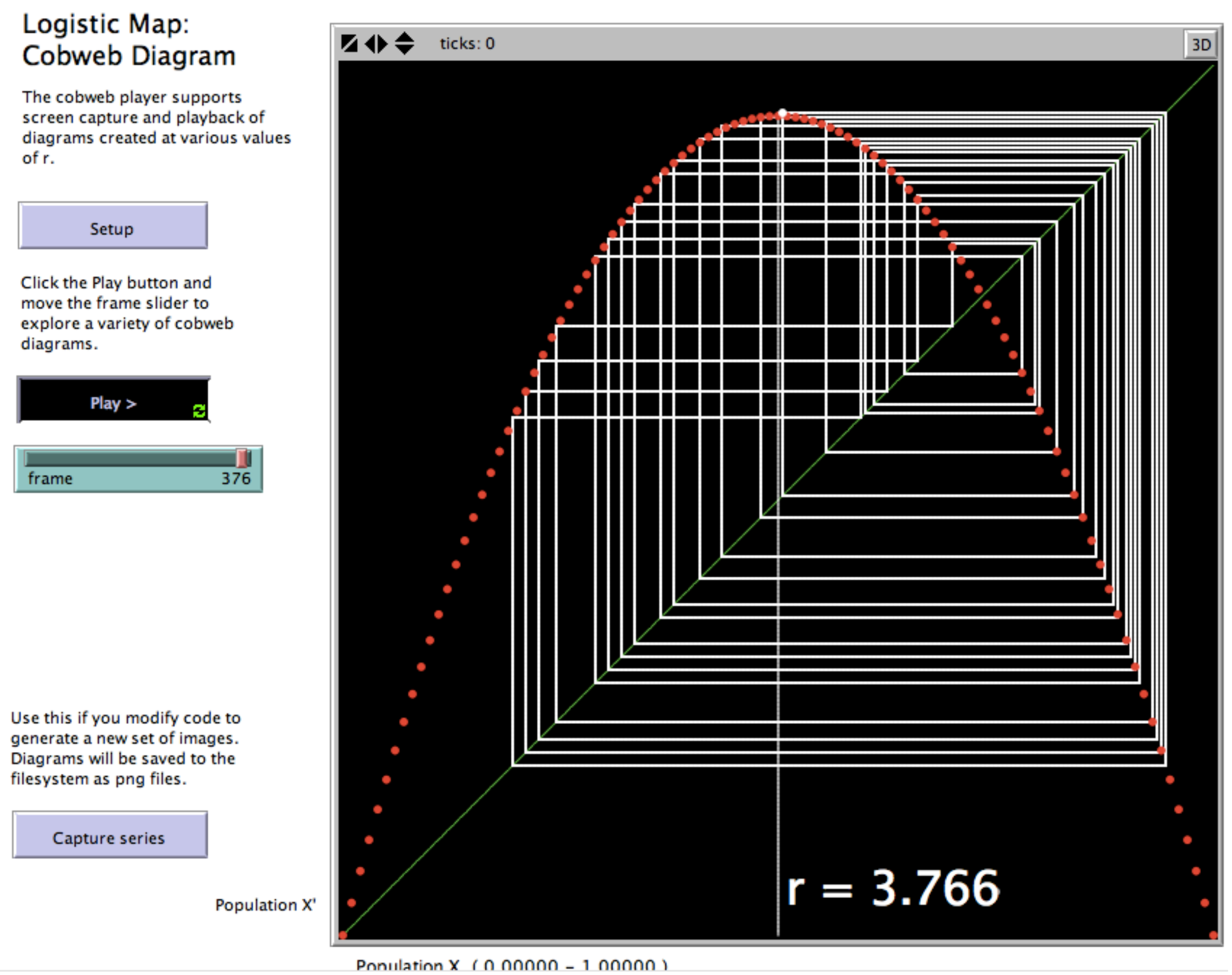
The NetLogo model included allows a student to explore iterations at various r values quickly. But the model also runs the risk of creating the illusion that the patterns are a consequence of some mysterious computer generated process. It is important to emphasize that it is *the function: its parameter and initial conditions* -- that is responsible for the patterns observed.



Assumes a starting population that is 50% of max.

r is the Malthusian parameter. Think of it as the fertility parameter--higher values of r mean more offspring per season.

A second model, Cobweb Player allows students to record and “flip” through images of the results of many iterations at each of 200 unique values of r. The resulting animation makes it clear there are distinct kind of behaviors that occur in different ranges of r.


Logistic Map:
Cobweb Diagram

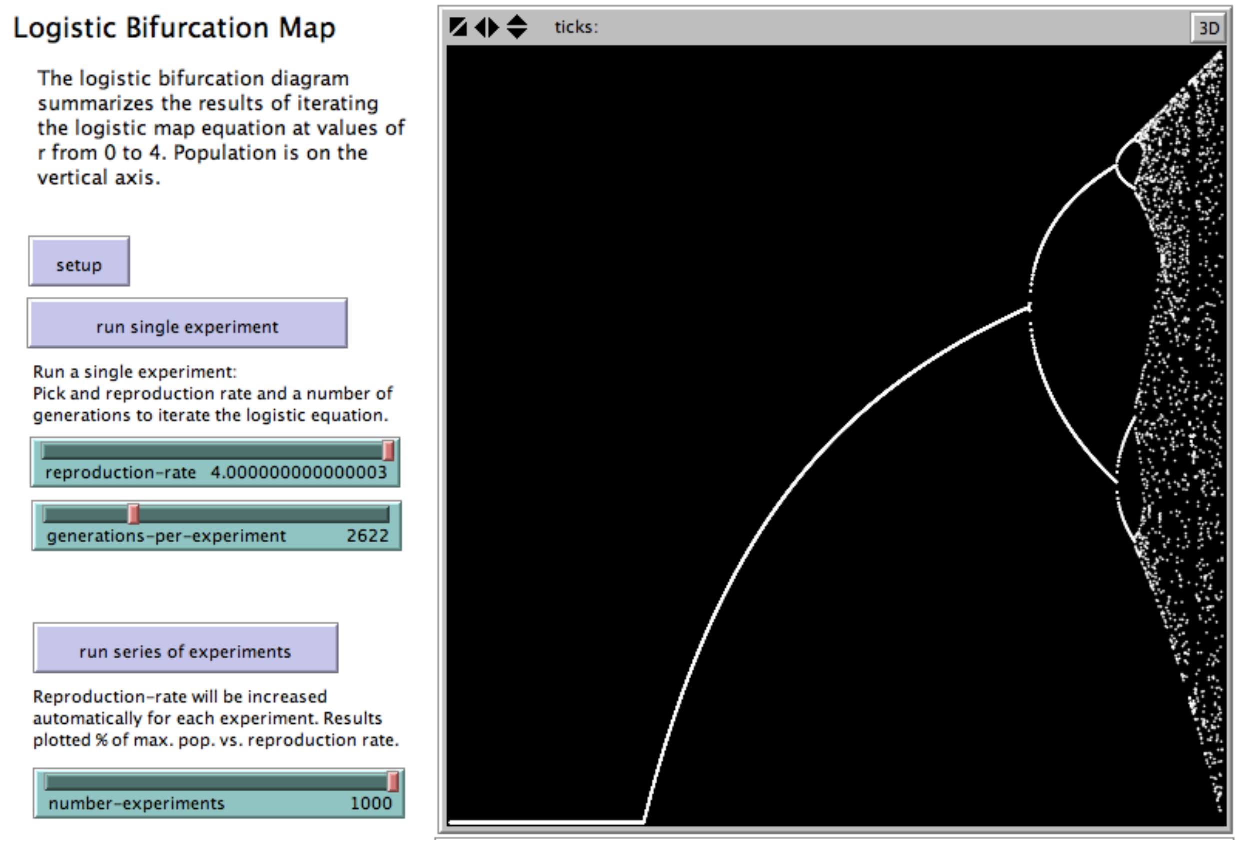
The cobweb player supports screen capture and playback of diagrams created at various values of r.

Click the Play button and move the frame slider to explore a variety of cobweb diagrams.

Use this if you modify code to generate a new set of images. Diagrams will be saved to the filesystem as png files.

To Create the Logistic Map

The final models in this series support visualization of ALL of the values approached by iterating the logistic map as we sweep through increasing values of r.



Logistic Bifurcation Map

The logistic bifurcation diagram summarizes the results of iterating the logistic map equation at values of r from 0 to 4. Population is on the vertical axis.

Run a single experiment:
Pick and reproduction rate and a number of generations to iterate the logistic equation.

reproduction-rate 4.000000000000003

generations-per-experiment 2622

Reproduction-rate will be increased automatically for each experiment. Results plotted % of max. pop. vs. reproduction rate.

number-experiments 1000

When r is varied over a range of 0-4 and the equation is iterated many times for each value of r, a plot called the **logistic map (or bifurcation diagram)** arises. This plot summarizes all the behavior of the logistic equation for different values of r.

The bifurcation diagram is created by choosing each value of r, and a given initial value of x. Using these two values, the difference equation is iterated for many, many times (10,000 +). The last 500-1000 values are captured and plotted against the value of r that was used in the equation. Initially, at low values of r, the final values all land at the same value. At higher values of r, the values land at multiple locations. Intricate patterns become apparent as the equation sweeps through different values of r.

The models provided allows students to zoom in on particular regions and vary the resolution (the level of iteration) at any particular point.

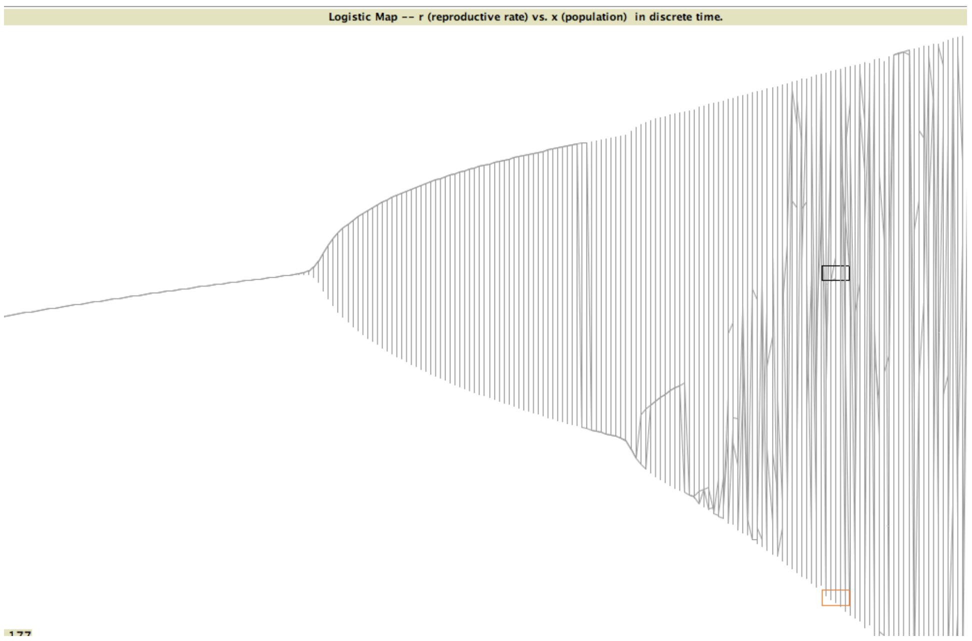
There are many interesting things to discover in with values of r between 3 and 4. As the value of r increase the number of values that the equation converges toward -- the attractors increases. These doublings become increasingly common but the rate of increased doubling itself follows a pattern. The Feigenbaum constant δ = 4.669... is a universal constant, like pi and e, describes the ratio (int the limit) between the value of range of r values for one bifurcation interval and the next. Discovered in 1978, the Feigenbaum constant holds for all one dimensional chaotic systems with a single quadratic maximum (think parabola-like shape). This *period doubling-cascade* is characteristic of chaotic systems and one of the sources of order that exist in the midst of apparent chaos.

There are other *islands of stability* at higher values of r where the number of attractors suddenly drops to (for example) 3. This behavior is counterintuitive but practically relevant in study of semiconductor properties and material science.

One area of special interest, called the *Pomeau–Manneville scenario, (* r varies from approximately 3.5699 to approximately 3.8284) is characterized by a periodic or laminar phase interrupted by bursts of aperiodic behavior.

Sensitivity to Initial Conditions

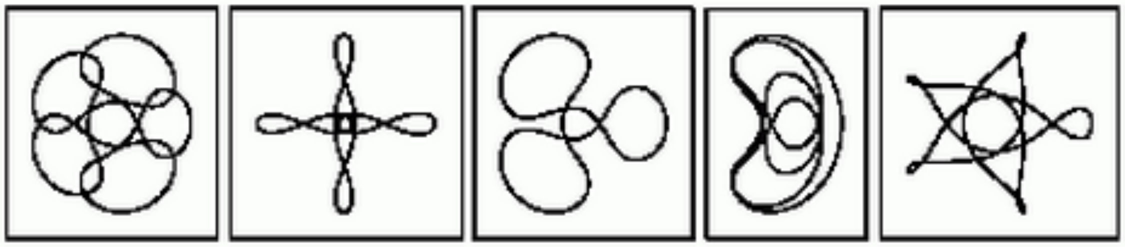
It seems quite reasonable to assume that things that are very similar suffer the same fates if exposed to similar processes. The study of chaotic systems reveals that intuition to be incorrect in certain situations. In this model we start with two values that are very close and use them as parameters--initial populations--in the logistic map. We see that after just a few dozens of iterations that calculated values may have diverged significantly. Furthermore, it is entirely possible that two values “more different in value” than the pair that we just considered could actually converge to closer values. Initial conditions come to mean less than the dynamics--the attractors--of the system.



The initial observation that some systems exhibited sensitivity to initial conditions is credited to the French mathematician Henri Poincare, (1854 – 1912). Poincare worked on a famous problem posed much earlier by Newton called the three body problem. The three body problem ask what the future motion of three mutually interacting bodies will be give their initial positions, masses and velocities at some particular point in time. The bodies are assumed to be interacting in accordance with the laws of classical mechanics.

As Poincare applied himself to this problem, he discovered that it was not possible to analytically describe (in terms of integral quantities such as velocities, position or mass ratios) the behavior of the three bodies in question. However, it *was* possible to describe the characteristic kinds of behavior that might occur. Poincare developed a new branch of mathematics called topology which formalized these concepts.

The pictures below show some of the possible repetitive orbits of an idealized planet moving in the plane of a pair of stars that are in a perfect elliptical orbit.



http://www.wolframscience.com/reference/notes/972d

Poincare also observed that small changes to the initial conditions might result in significant changes to the overall behavior of the system. For this insight, Poincare is recognized as one of the founders of the chaos theory.

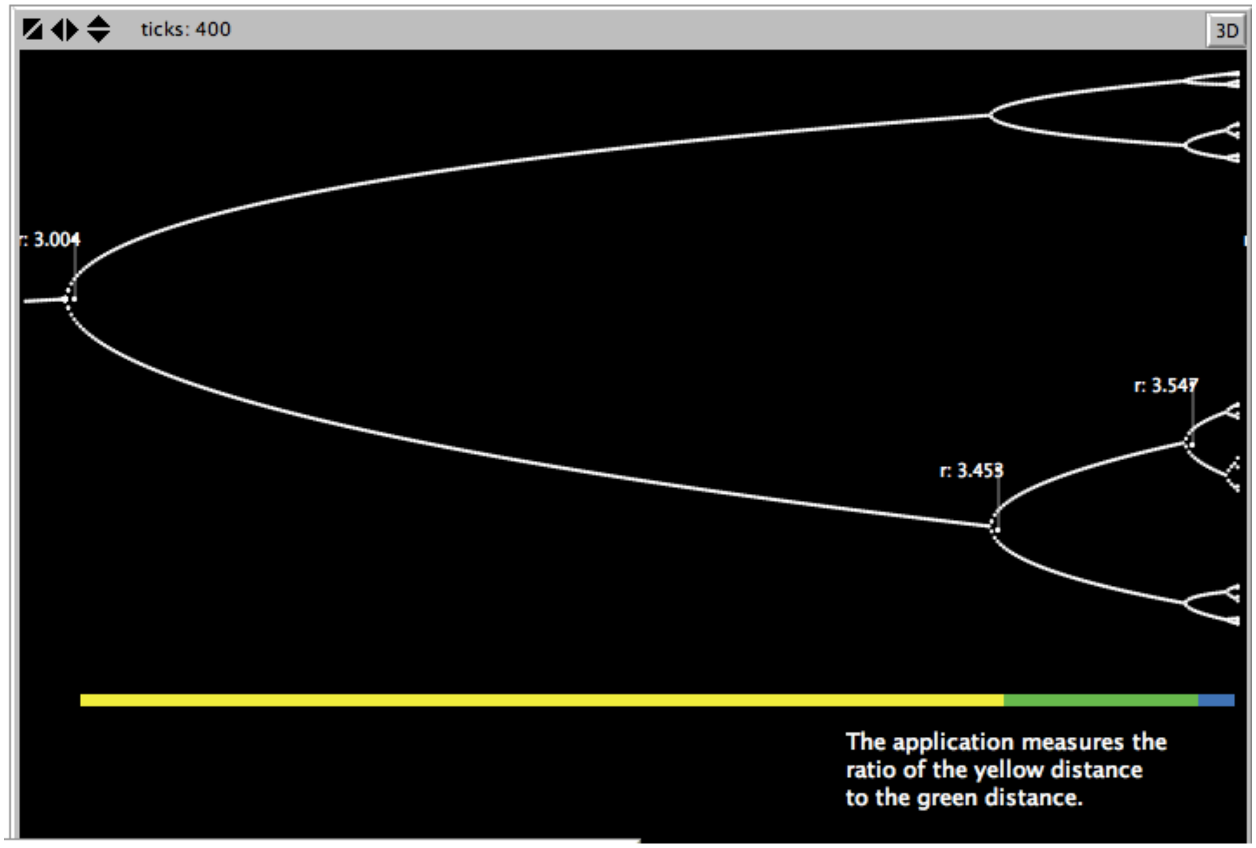
Universal Constants

The logistic map features a series of bifurcations at increasingly smaller intervals as the parameter r increases. It is natural to wonder if there might not be some pattern underlying branching pattern. One of many objectives of science is to identify stable regularities underlying complex phenomena. The phenomena in question here is somewhat artificial--the increasingly less well organized plotted points on the logistic map--but the fact that a regularity can be found is quite remarkable. The further fact, that analogous regularities are found in physical systems such as the change from laminar to turbulent flow (in hydraulic or pneumatic systems) are even more remarkable.

In 1975 Mitchell Feigenbaum discovered two universal constants for functions approaching chaos via period doubling. That is, Feigenbaum noticed that many systems move from a single stable point (an attractor), through a series of periodic doubling, finally to a state of aperiodic “randomness”. Those systems which move through a period doubling phase have in common the fact that the doubling proceeds at an increasing rate.

It was surprising to discover that the rate of increase proceeded at a specific, single rate. Fiegenbaum and his colleagues were astonished, however, when they observed that the same rate applied to very different functions undergoing iteration and bifurcation. Apart from the astonishing universality of the constant, there is a practical benefit to be derived from applying the constant to the first few bifurcation in any system.   Noting the ratios between the first few bifurcations supports an accurate estimate of point at which a system becomes chaotic. Knowledge of this point might be useful to someone interested in managing a physical systems.

The final model of the logistic equation/ logistic map series present a straightforward way to visualize the ratios involved in determining Feigenbaum's constant.



The application measures the ratio of the yellow distance to the green distance.

Flake, Gary (1998) “The Computational Beauty of Nature” MIT Press

Feigenbaum, Mitchell. (1980) “Universal Behavior of Nonlinear Systems“ LOS ALAMOS SCIENCE 1 1 1980

Kaldor, Nicholas (1934), "A Classificatory Note on the Determinateness of Equilibrium" The Review of Economic Studies, 1, 122-136.

Chaos: An Introduction to Dynamical Systems, K.T. Alligood, T.D. Sauer, J.A. Yorke, Textbooks in mathematical sciences ,Springer, 1996, ISBN 978-0-38794-677-1

Historical Sidebar:

During the 1970's and 1980's a theory based on the logistic equation was advanced and received widespread attention. The theory claimed that organisms evolved to adopt reproductive life strategies that capitalized on one or the other of the two antagonistic terms in the logistic equation. Organisms with r-selective strategies capitalized on the first term in the equation by emphasizing high-fecundity, early maturity, short generation time and the ability to disperse widely.

On the other hand, other organisms were classified as K-selective strategists. These organisms tended to be larger, more long-lived, had high parental investment in offspring and employed strategies that addressed the problems of running into the upper limits of population growth. Unfortunately, like so many beautiful theories with intuitive appeal, the r/K selection theory failed when subjected to empirical testing--examples of long-lived, large but also high fecund organisms (trees) upset the theoretical simplicity that the r/K selection theory had suggested.