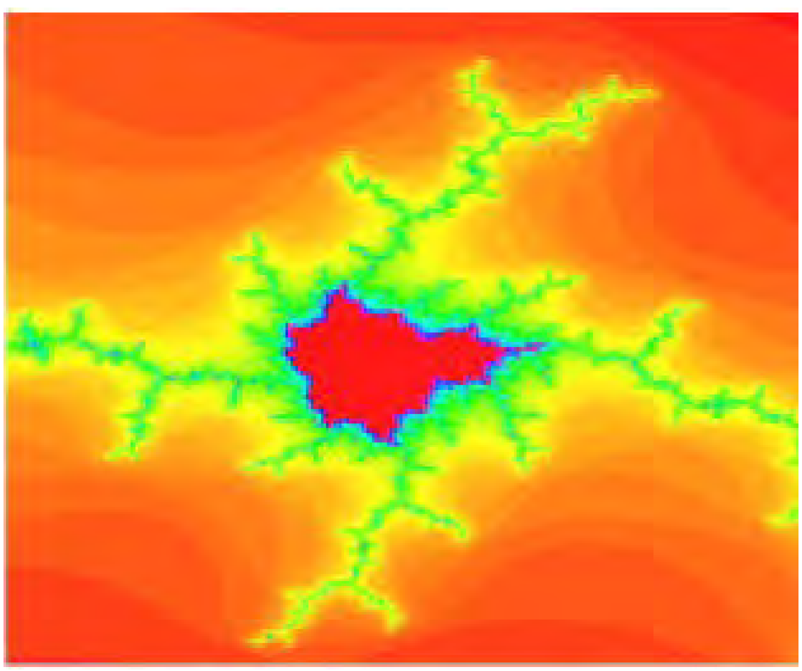
Chaotic Rendezvous:

Introductory Chaos Experimentation Using Maple Software and a Desktop Computer

by Thomas E. Oberst



Chaotic Rendezvous: Introductory Chaos Experimentation Using Maple Software and a Desktop Computer

Thomas E. Oberst

Duquesne University
Department of Physics
Pittsburgh, PA 15282

Chapter 7

The Orbit Diagram

Up to this point, our exploration of orbits has been limited to one orbit at a time—whether such exploration was done with a cobweb diagram or a point-plot, or with just a straight list of numerical values. We have always chosen one particular function to explore, one initial seed to begin with, and gone about our merry experimentation. Imagine, however, if we could think of a way to explore several functions at once. What if we could map out the entire global behavior of *all* members of a particular family of functions: such as the entire logistic family, or the entire quadratic family?

To come up with such a comprehensive exploration scheme, we must think back to what were the most important qualities among the orbits explored thus far in this manual. Many options lie before us as to what properties to look at and what peculiarities to emphasize in orbits, but we must choose those that are most telling. It was never very important, for example, in what order points fell during a particular orbit. What was most important was always *how many* points existed in a particular orbit. I.e., is the orbit fixed, does it converge to a single fixed value, is it a 2-cycle, etc? And for a given family of functions (most of our experiments to this point have involved only the logistic and quadratic families), how does this number of points change as we vary the parameter of the family (i.e., λ for the logistic family and c for the quadratic family)? It would make sense, then, to create a graph showing iterate values versus the parameter value of the function family. Such a graph would display the behavior of the orbits of the members of the family as the parameter is continuously varied.

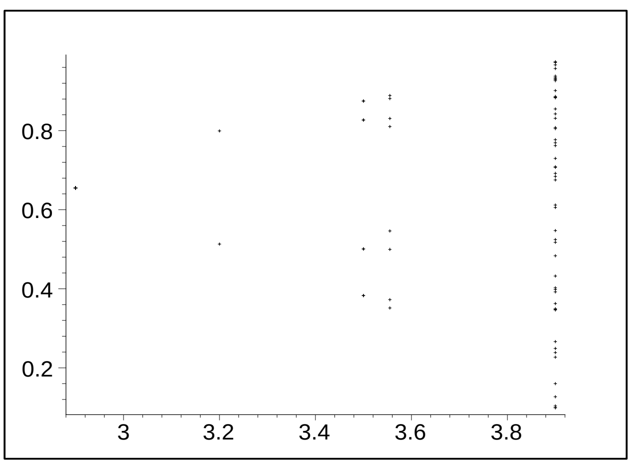
Let's try creating such a graph for our most familiar family of functions: the logistic family, $F_\lambda(x) = \lambda x(1-x)$. First we define the function:

```
> f:=unapply(lambda*x*(1-x), x, lambda);
```

$$f := (x, \lambda) \to \lambda\, x\, (1-x)$$

Now we wish to plot values of λ (that is, various logistic functions) on the abscissa axis and the values in the orbits of each of these functions on the ordinate axis. These iterate values are to be plotted directly above their respective λ-value, almost like a histogram, except that the iterate values will be placed exactly at their values according to the ordinate scale, rather than just being piled on top of each other as in a histogram. To accomplish this, we will first have to choose several lambda values to use in our plot, and then we will have to compute the orbit for each of these functions. For diversity, let's choose one logistic function in which orbits converge to a single fixed point ($\lambda = 2.9$), one logistic function with a 2-cycle ($\lambda = 3.2$), one logistic function with a 4-cycle ($\lambda = 3.5$), one with an 8-cycle ($\lambda = 3.555$), and one logistic function from the chaotic region of the family ($\lambda = 3.9$). We define the arrays $X1$ through $X5$ to hold the iterate values for these five functions, and we begin the orbit of each of these functions with the point $x_0 = 0.5$ (recall that for initial seeds chosen outside of the closed interval [0,1], all orbits diverge to $-\infty$ for the logistic map when $0 \leq \lambda \leq 1$). For each orbit, we calculate 100 iterations and throw out the first 50 in order to show the asymptotic behavior of the orbits.

```
> X1:=array(0..100): X2:=array(0..100): X3:=array(0..100):
> X4:=array(0..100): X5:=array(0..100):
> X1[0]:=.5: X2[0]:=.5: X3[0]:=.5: X4[0]:=.5: X5[0]:=.5:
> for j from 1 to 100 do X1[j]:=f(X1[j-1],2.9) od:
> for j from 1 to 100 do X2[j]:=f(X2[j-1],3.2) od:
> for j from 1 to 100 do X3[j]:=f(X3[j-1],3.5) od:
> for j from 1 to 100 do X4[j]:=f(X4[j-1],3.555) od:
> for j from 1 to 100 do X5[j]:=f(X5[j-1],3.9) od:
> with(plots):
> pointplot( {seq([2.9,X1[i]],i=50..100),seq([3.2,X2[i]],i=50..100),
seq([3.5,X3[i] ],i=50..100),seq([3.555,X4[i]],i=50..100),seq([3.9,X5[i]],
i=50..100) });
```



The diagram we have created here is called an *orbit diagram*. The particular diagram shown above, while crude, is informative. It tells us (as was stated above) that when the parameter of the logistic family is 2.9 the orbit converges to a single fixed value; when the parameter is 3.2 the orbit converges to a 2-cycle; when the parameter is 3.5 the orbit is attracted to a 4-cycle; at 3.555 we see attraction to an 8-cycle; and, finally, at 3.9 we get a chaotic orbit. More importantly, the diagram suggests that the transition from a single fixed point orbit to a chaotic orbit is a continuous function of λ. To be absolutely sure that this is the case we would have to fill in many many more functions along the x-axis. Creating all the arrays, defining the initial seed of each array, and calculating all the orbits seems like a horrific task. This is especially true when we don't know what to expect from a particular function. For the logistic family, we already knew that interesting things happen for $1 \leq \lambda \leq 4$. Without such knowledge, we could have spent many futile hours creating orbit diagrams for logistic functions whose parameters lie elsewhere among the real numbers.

To make things easier, I have designed a procedure (shown below) which, given an interval of parameter values, subdivides that interval into a number of parameter values, computes the orbit for each of these values, throws out the transient values in each orbit, and then creates an orbit diagram. Like the `cobweb` procedure introduced in Chapter 5, the procedure must be typed into your Maple interface verbatim as it appears below. I also suggest that you save this procedure as a Maple worksheet so that it will be available to you for future use. I created this procedure to save you time and make orbit diagrams an efficient way of exploring families of functions, not to encourage rote. I encourage you to read through the procedure carefully and understand its operations. The Maple entries used to create the crude orbit diagram above were the inspiration for this procedure.

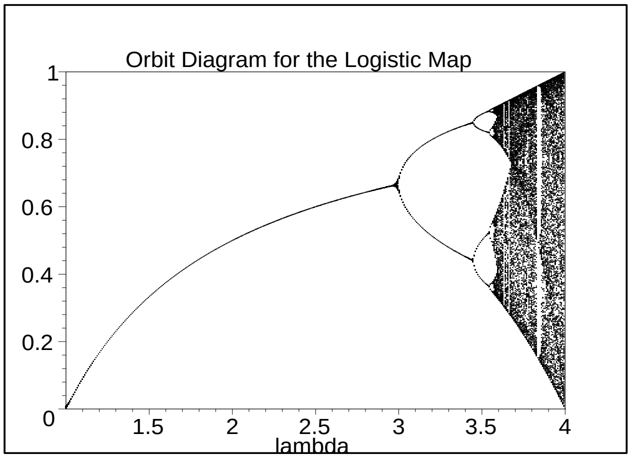
```
> orbit:=proc(f_of_x, parameter, functions, x0, iterations, throwout,
> parameter_min, parameter_max, iterate_value_min, iterate_value_max, Symbol)
> local f, X, epsilon, i, j:
> with(plots):
> f:=unapply(f_of_x, x, parameter):
> epsilon:=(parameter_max-parameter_min)/functions:
> X:=array(0..functions, 0..iterations):
> for i from 0 to functions do X[i,0]:=x0 od:
> for i from 0 to functions do for j from 1 to iterations do
> X[i,j]:=evalf(f(X[i,j-1], parameter_min+i*epsilon)) od od:
> pointplot({seq(seq([parameter_min+i*epsilon,X[i,j]], i=0..functions),
> j=throwout..iterations)}, symbol=Symbol, scaling=unconstrained, view=
> [parameter_min...parameter_max, iterate_value_min...iterate_value_max],
> axes=boxed, labels=[parameter,``]);
> end proc;
```

$orbit :=$ **proc**$(f\_of\_x,\ parameter,\ functions,\ x0,\ iterations,\ throwout,\ parameter\_min,$
$parameter\_max,\ iterate\_value\_min,\ iterate\_value\_max,\ Symbol)$
local $f,\ X,\ \varepsilon,\ i,\ j;$
with$(plots)$;
$f :=$ unapply$(f_of_x,\ x,\ parameter)$;
$\varepsilon := (parameter\_max - parameter\_min)/functions$;
$X :=$ array$(0..functions,\ 0..iterations)$;
for i **from** 0 **to** $functions$ **do** $X_{i,0} := x0$ **end do** ;
for i **from** 0 **to** $functions$ **do**
for j **to** $iterations$ **do** $X_{i,j} :=$ evalf$(f(X_{i,j-1},\ parameter_min + i * \varepsilon))$ **end do**
end do;
pointplot$(\{$seq(seq$([parameter_min + i * \varepsilon,\ X_{i,j}],\ i = 0..functions),$
$j = throwout..iterations)\},\ symbol = Symbol,\ scaling = unconstrained, view =$
$[parameter\_min..parameter\_max,\ iterate\_value\_min..iterate\_value\_max],$
$axes = boxed,\ labels = [parameter,\ ``])$
end proc

This procedure creates a new command called `orbit`. The `orbit` command takes 11 arguments. The first argument is the family of functions for which the orbit diagram is to be created. This family must be expressed in terms of x (must be of the form $f(x)$). The second argument is the parameter of the function. If, for example, you enter the quadratic family as the first argument in the `orbit` command, $x^2 + c$, the second argument of the `orbit` command must simply be the letter c. The third argument is the number of functions you wish to compute orbits for—or, in other words, the number of parameter values you wish to plot along the x-axis of your orbit diagram. The fourth argument is the initial seed at which to begin the iteration process for each of your functions. The fifth argument specifies the number of iterations you wish to perform for each of your functions. The sixth argument specifies how many of these iterations you wish to throw out for each function before creating your diagram (in order to achieve asymptotic orbits). The seventh and eighth arguments respectively specify the maximum and minimum parameter values (maximum and minimum x-axis values) to be used for the diagram. The ninth and tenth arguments respectively specify the maximum and minimum iteration values (maximum and minimum y-axis values) to be displayed in the diagram. And finally, the eleventh argument allows you to choose what type of symbol to use to plot the points in your orbit diagram. Your choices for this argument are point, cross, diamond, circle, and box.

A more complete orbit diagram for the logistic map is shown below. In this orbit diagram we have plotted 400 λ-values along the x-axis. For each of the logistic functions corresponding to these 400 λ-values we have calculated the first 400 points in the orbit of $x_0 = 0.5$. Since we are interested only in the asymptotic orbits of these functions, the first hundred points were thrown out. Using our prior knowledge that everything interesting happens between $1.0 \leq \lambda \leq 4.0$ (see Table 4.1), we choose 1.0 and 4.0 as our maximum and minimum x-values. We also know from prior investigations that we must choose our initial seed from the interval [0,1], and that the values in such an orbit will never leave the interval [0,1]. Hence we set $x_0 = .05$ and choose our maximum and minimum y-values to be 0 and 1:

```
> orbit(x*lambda*(1-x), lambda, 400, .5, 400, 100, 1, 4, 0, 1,
point)
```

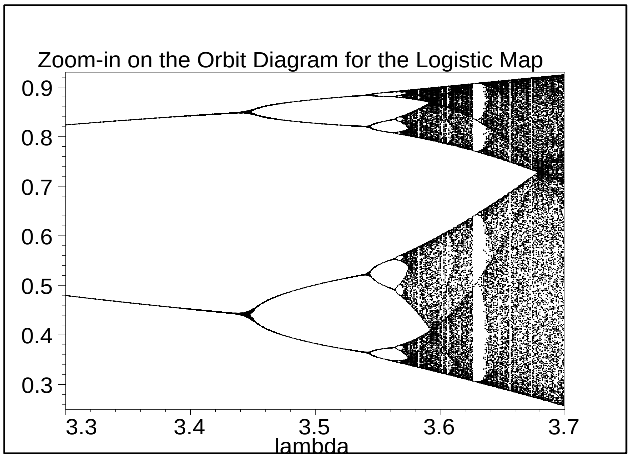


This remarkable diagram provides an overview of all of the dynamics of the logistic family between $1.0 \leq \lambda \leq 4.0$ (as described in Table 4.1). For all logistic functions whose parameters fall between $\lambda = 1.0$ and $\lambda = 3.0$ we see that the asymptotic orbit consists of a single point. The orbit diagram conveniently lets us determine this value for a given parameter by using the scale on the y-axis. For example, if we move vertically upwards from the point $\lambda = 2.5$ on the x-axis until we reach the curve in the orbit diagram, and then move horizontally to the left, we see that the orbit of $x_0 = 0.5$ for the function $F_{2.5}(x) = 2.5x(1-x)$ approaches a value of about 0.6 (0.6 can be shown to precisely be an attracting fixed point of the function $F_{2.5}(x)$ using analytical methods). From $\lambda = 3.0$ until $\lambda \approx 3.45$, we see the orbits become 2-cycles. At $\lambda \approx 3.45$, the orbits become 4-cycles. At $\lambda \approx 3.54$, the orbits become 8-cycles. Then 16-cycles. Then 32-cycles. And so on. The orbit diagram allows us to truly see how the transition to chaos takes place through this series of *period doublings*. This series of period doublings is often called the *period-doubling route to chaos*.

Recall for a moment the task set before you in Experiment 1.3, when you were asked to compute numerical lists of the iterate values in orbits for the logistic map for 10 to 20 λ-values, determine the fate of such orbits (whether they were fixed, eventually fixed, 2-cycles, etc.), and then draw conclusions about the global dynamics of the family. I don't think I need to point out the appreciation that should be shown for the orbit diagram. Orbit diagrams contain a plethora of information, and are arguably the most useful diagrams in the study of dynamical systems.

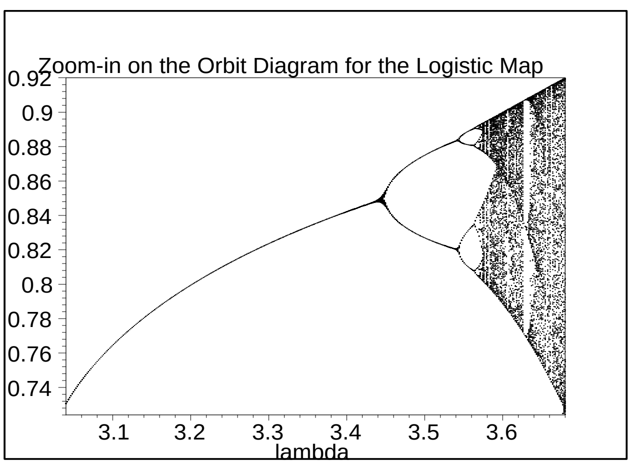
To take a closer look at the period-doubling route to chaos, let's zoom in on the region of the logistic map's orbit diagram stretching from $\lambda = 3.3$ to $\lambda = 3.7$ along the x-axis and from 0.25 to 0.93 along the y-axis (the "upper-half" of the orbit diagram):

```
>  orbit(x*lambda*(1-x), lambda, 400, .5, 400, 100, 3.3, 3.7, .25, .93,
point);
```



We then zoom in again on the "upper half" of this picture:

```
>  orbit(x*lambda*(1-x), lambda, 400, .5, 400, 100, 3.04, 3.68, .724, .92,
point);
```



Note the remarkable similarity between the zoom-in of the region [3.04...3.68,0.724...0.920] shown here and the original entire orbit diagram above! The long single-valued curve between $\lambda = 3.04$ and $\lambda \approx 3.44$ in this zoom-in is actually the greatest value in the 4-cycle region of the logistic family. And the first period doubling curves between $\lambda \approx 3.44$ and $\lambda \approx 3.54$ in this zoom-in actually represent the two greatest values in the 8-cycle region of the logistic family. Ignoring these facts, however, the general shape and structure of the two pictures are identical. The period-doubling route to chaos is also of the same structure. Evidently, this orbit diagram is a fractal! (Recall from Chapter 2 that fractals are geometric objects which show self-similarity at every level of magnification.) If we were to zoom in again and focus on the region [3.48...3.59,0.85..0.90], we would again see a picture which looks very much like the one shown here as well as the original complete orbit diagram.

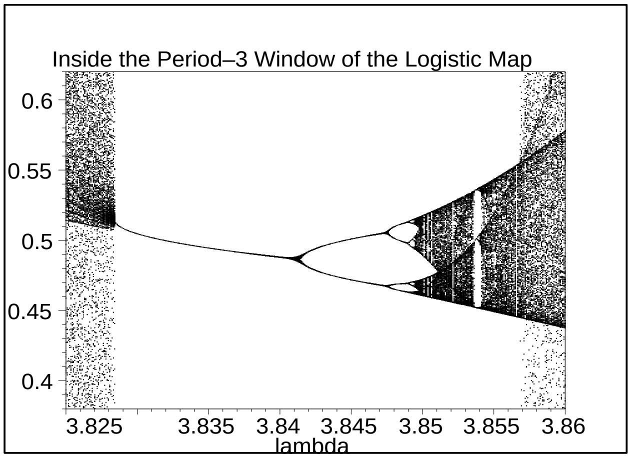
In the last orbit diagram zoom-in shown above, as well as in the original complete orbit diagram for the logistic map, note the small vertical white space—or "window"—that opens in the middle of the chaotic region. In the original map this window appears around $3.828 < \lambda < 3.857$. In the zoom-in this window appears around $3.625 < \lambda < 3.629$. If you look closely at these regions, you will see that they are not empty. Three small dark lines traverse the window, indicating that periodic points with prime period three are present (a 3-cycle). For this reason the space is called the *period-3 window*.

Knowledge that a function has an orbit of period three tells us much more about the function than just that it has an orbit of period three. The famous Li-Yorke theorem states: If $F : \Re \rightarrow \Re$ is continuous and F has a periodic point of prime period three, then F also has periodic points of all other periods ($\Re$ represents the set of all real numbers). This theorem was published in 1975 in a landmark paper entitled "Period Three Implies Chaos." This paper marks the first use of the word "chaos" in scientific literature. The theorem itself is remarkable. Just by finding a periodic point of prime period three, we are ensured that the function has periodic points of all other periods—including points of periodicity ∞, a.k.a. chaotic points!

But if periodic points of all other periods are present, why does the period-3 window appear to be so empty in the orbit diagram? This question brings up another important attribute of the orbit diagram: In an orbit diagram, only attracting points and attracting cycles are displayed. Thus, in the period-3 window, only the 3-cycle is attracting. Periodic points of all other periodicities are present, but they must all be repelling.

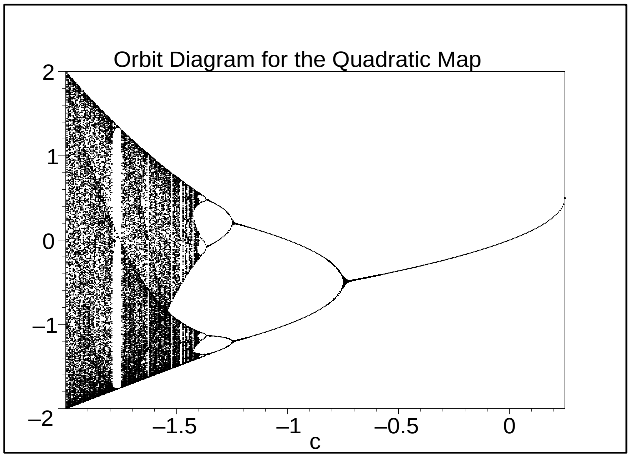
The next figure shown below is a zoom-in on one of the points of period three in the period-3 window of the orbit diagram for the logistic map. This picture reveals that the 3-cycle extends only from about $\lambda = 3.828$ to $\lambda = 3.841$, where it gives way to a period-doubling route to chaos. Note again the truly amazing self-similarity that this small portion of the orbit diagram shows to the complete orbit diagram.

```
> orbit(x*lambda*(1-x), lambda, 400, .5, 700, 100, 3.825, 3.86, .38, .62,
point);
```



The general shape of this orbit diagram is not unique to the logistic map. If we were to construct an orbit diagram for, say, the quadratic map, we would achieve a similar-looking figure (shown below). This remarkable universality of the orbit diagram applies to all unimodal functions—i.e., continuous functions with a single local extremum. The logistic map and quadratic map are both unimodal. In 1973, Metropolis and a few of his colleagues published a paper which studied the dynamical properties of all unimodal maps of the form $x_{n+1} = rf(x_n)$, where f(x) also satisfies $f(0) = f(1) = 0$. The paper contains a powerful theorem which proves that as r is varied, the order in which periodic orbits appear is entirely independent of $f(x)$. Only the general unimodal shape of the map matters! The universal sequence in which the periodic orbits appear is now known as the famous *U-sequence*. The U-sequence begins by moving from a single fixed point into the period-doubling cascade, where the periodic orbits appear in the order: 2, 4, 8, 16, ..., 2^n. Later the U-sequence includes periodic orbits of period 6, 5, and 3. From the period-3 window, period doubling takes over again. This time, however, the periodic orbits resulting from the period doubling are: 6, 12, 24, 48, ..., $2^n \cdot 3$.

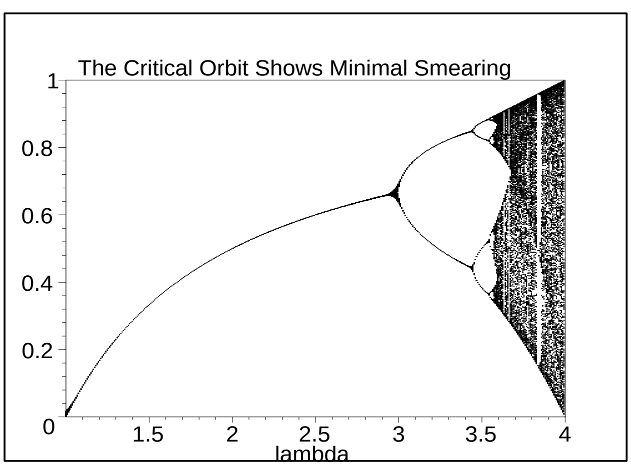
```
>  orbit(x^2+c, c, 400, 0, 400, 100, -2, 1/4, -2, 2,
point);
```



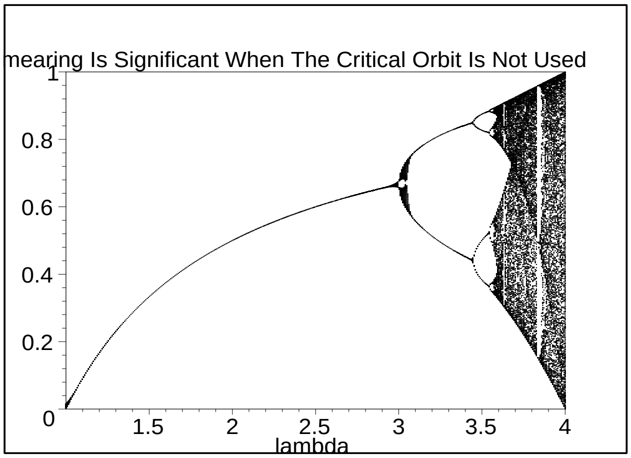
Also important to the orbit diagram is the initial seed chosen to compute the orbits in the diagram. In the example above with the logistic map, the initial seed $x_0 = 0.5$ was used in all cases. We know from previous experiments that the asymptotic orbit of the logistic map is the same regardless of what initial seed we choose, provided that it is chosen from the non-diverging open interval (0,1). (Except in the special case when $\lambda = 4$ and $x = 0.5$ becomes an eventually fixed point. In such situations 0.5 should be avoided as an initial seed.) Some initial seeds from within (0,1) take a longer time to reach their asymptotic orbits than others. Hence, if different initial seeds are used to create several orbit diagrams and exactly 100 points are thrown out each time, a smearing effect will be seen in those orbit diagrams whose initial seeds took longer to reach their asymptotic orbits.

In the two orbit diagrams for the logistic map below I have thrown out only 50 points, so as to accentuate the smearing effect. The first orbit diagram again uses the initial seed $x_0 = 0.5$. The second diagram uses the initial seed $x_0 = .\overline{123}$:

```
> orbit(lambda*x*(1-x), lambda, 400, .5, 400, 50, 1, 4, 0, 1,
point);
```



```
> orbit(lambda*x*(1-x), lambda, 400, .123123123123, 400, 50, 1, 4, 0, 1,
point);
```



Notice the "smeared" regions near the first period doubling ($\lambda = 3.0$) in both orbit diagrams. However, the smearing effect is much more severe in the second of the two diagrams. To avoid this undesirable smearing effect, we always use the *critical point* of a function to calculate the orbits in its orbit diagram. The critical point x_0 of a function f is defined as the point for which $f'(x_0) = 0$. For our unimodal maps, the critical point is the same as the local extremum. For the logistic map this value is 0.5, and for the quadratic map this value is 0. The reason for using the orbit of the critical point (often called the *critical orbit*) to create our orbit diagrams is that if an attracting point or attracting cycle exists for the function, we can be sure that the critical point will "find" it.[1]

7.1 Experiment: Orbit Diagrams for the Quadratic Map

1. Create an orbit diagram for the quadratic map: $Q_c(x) = x^2 + c$. Display only the parameter range where interesting things happen. (By interesting things, I mean fixed and periodic orbits and the period-doubling route to chaos.) Also be sure to display only the domain into which the iterate values fall. Use the critical orbit $x_0 = 0$ to compute all orbits in the diagram. Don't forget to throw out the first 100 or so points in each orbit so that you are plotting only the asymptotic orbits.

   Note: Creating an orbit diagram involves the calculation of a very large number of points. Unless you are very ambitious or absolutely need to see greater detail, I recommend that you keep the number of parameters below 500 and the number of iterations below 500. The actual number of points computed by Maple is (Parameters

[1]Proof of this fact can be found in Devaney, Robert L., A First Course in Chaotic Dynamical Systems. Addison-Wesley Publishing Co., 1992. Page 158.

× Iterations). The actual number of points displayed is (Parameters × (Iterations-Throwouts)). For most of the orbit diagrams displayed in this chapter, 400 parameter values and 400 iterations were used. Therefore, Maple calculated 160,000 points for each of these diagrams! In the diagrams for which we chose to throw out the first 100 iterations, Maple plotted 120,000 points! The time required to carry out such calculations and produce such plots depends on the speed of your computer. These graphs took me an average of 3-4 minutes each to produce. If less detail is required, you should use around 200 to 300 parameter values and 200 to 300 iterations. When first producing an orbit diagram, I recommend that you use only about 100 parameter values and 100 iterations (or fewer), since you often will not know if you are in the proper parameter and/or iterate value domain (i.e., x and y-value ranges). Once you have found a desired area, you can up your number of parameter values and number of iterations.

2. Create three (or more) successive magnifications of the orbit diagram for the quadratic map in order to show the fractal nature of the diagram. Adjust your parameter range and iterate value domain so that each magnification resembles the original orbit diagram.

3. Zoom in on your orbit diagram and take a closer look at the period-3 window. In what parameter range does the period-3 window fall? Also find the largest period-5 window and period-6 window in your orbit diagram. What are the parameter ranges of these windows?

4. Create a new orbit diagram for the quadratic map, this time throwing out only the first 10 or fewer iterations of each orbit. How is the diagram different from your first one? Are smearing effects present? Explain where and why this smearing takes place.

5. Print your results and responses from the steps above and prepare them to be turned in to your instructor.

7.2 Experiment: Orbit Diagrams for Other Functions

1. Create an orbit diagram for the family of functions

$$\lambda \sin x$$

where $\lambda \in \Re$. Start at λ-values around 0 and move out in either direction from there. Use the critical point nearest to 0 to calculate your orbits. The sin function is a repeating function. Is any of this repetitious nature apparent in the orbit diagram (i.e., can you find more than one orbit diagram)? Does this orbit diagram resemble those of the unimodal logistic and quadratic maps? Why or why not?

2. Create an orbit diagram for the family of functions

$$\lambda \sin^2 x$$

where $\lambda \in \Re$. Start at λ-values around 0 and move out in either direction from there. Use the critical point nearest to 0 to calculate your orbits. The sin function is a repeating function. Is any of this repetitious nature apparent in the orbit diagram (i.e., can you find more than one orbit diagram)? Does this orbit diagram resemble those of the unimodal logistic and quadratic maps? Why or why not?

3. Create an orbit diagram for the family of functions

$$\lambda \cos x$$

where $\lambda \in \Re$. Start at λ-values around 0 and move out in either direction from there. Use the critical point nearest to 0 to calculate your orbits. The cos function is a repeating function. Is any of this repetitious nature apparent in the orbit diagram (i.e., can you find more than one orbit diagram)? Does this orbit diagram resemble those of the unimodal logistic and quadratic maps? Why or why not?

4. Create an orbit diagram for the family of functions

$$\lambda \cos^2 x$$

where $\lambda \in \Re$. Start at λ-values around 0 and move out in either direction from there. Use the critical point nearest to 0 to calculate your orbits. The cos function is a repeating function. Is any of this repetitious nature apparent in the orbit diagram (i.e., can you find more than one orbit diagram)? Does this orbit diagram resemble those of the unimodal logistic and quadratic maps? Why or why not?

5. Create an orbit diagram for the family of functions

$$x^4 + c$$

where $c \in \Re$. Start at c-values around 0 and move out in either direction from there. Use the critical point nearest to 0 to calculate your orbits. This function is not unimodal. Does the orbit diagram resemble those of the unimodal maps? How is it the same? How is it different? Is the U-sequence the same?

6. Create an orbit diagram for the family of *piecewise* functions

$$T(x) = \begin{cases} cx & 0 \leq x < \frac{1}{2} \\ -cx + c & \frac{1}{2} \leq x < 1 \end{cases}$$

where $c \in \Re$. These functions are commonly called “tent maps.” Start at c-values around 0 and move out in either direction from there. This function is not continuous and therefore cannot be described as being unimodal. Does the orbit diagram resemble those of the unimodal functions described above? If so, why? If not, why not? Is the U-sequence the same?

7. Print your results and responses from the steps above and prepare them to be turned in to your instructor.