The Logistic Map

A *Mathematica* notebook written for Math 118: Dynamical Systems

Matthew Leingang, Course Assistant
9 March 1999

■ Definition

```
In[1]:= L[μ_, x_] := μ x (1 - x)

In[2]:= L_μ_[x_] := L[μ, x]

In[3]:= L_μ'[x]

Out[3]= (1 - x) μ - x μ

In[4]:= Simplify[%]

Out[4]= μ - 2 x μ
```

■ Orbit Structure, Algebraic

■ Fixed Points

L, being a quadratic in x, has two fixed points:

```
In[5]:= Solve[L[μ, x] == x, x]
```

Out[5]= $\{\{x \to 0\}, \{x \to \frac{-1+\mu}{\mu}\}\}$

To determine their stability, we take the derivatives at the fixed points.

```
In[6]:= L_μ'[x] /. %
```

Out[6]= $\{\mu, 1 - \mu + \left(1 - \frac{-1+\mu}{\mu}\right)\mu\}$

```
In[7]:= Simplify[%]

Out[7]= {μ, 2 - μ}
```

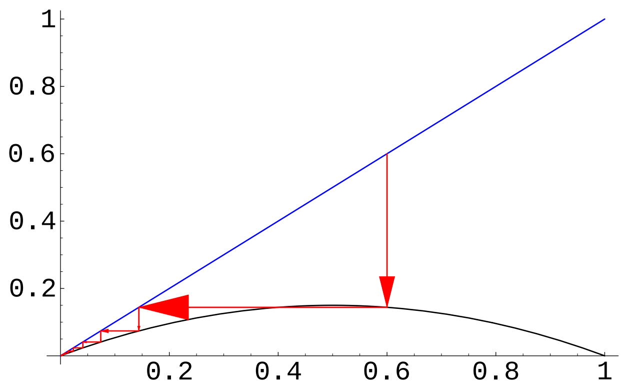
So we see that the fixed point 0 is stable as long as $0 < \mu < 1$, and the fixed point $p_\mu = 1 - 1/\mu$ is positive for $\mu > 1$ and stable when $1 < \mu < 3$. Let' s verify this graphically.

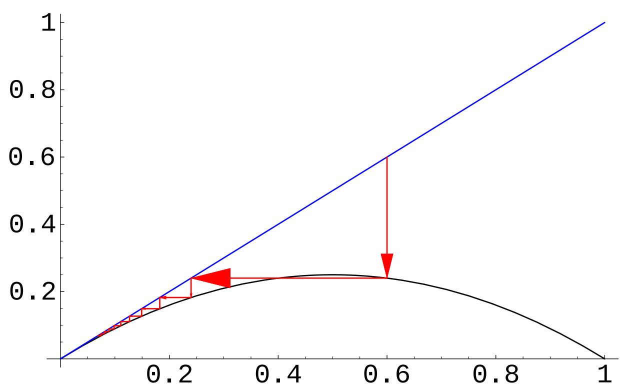
```
In[8]:= << Graphics`Arrow`
```

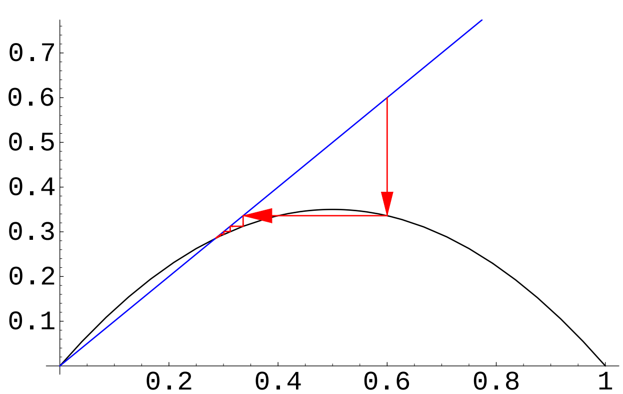
```
In[9]:= << Graphics`Colors`
```

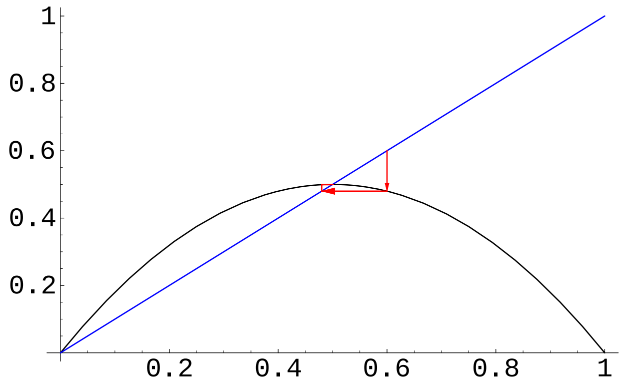
```
In[10]:= << "~/math/118/Bounce.m"
```

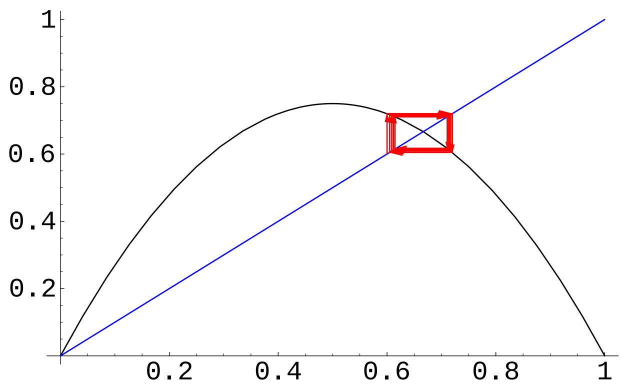
```
In[31]:= Plot[{L#[x], x}, {x, 0, 1},
          PlotStyle -> {{}, Blue},
          Epilog -> {Red,
             BouncePath[L#, 0.6, 10,
             HeadScaling -> Relative]}]& /@
         {0.6, 1, 1.4, 2, 3, 3.5}
```

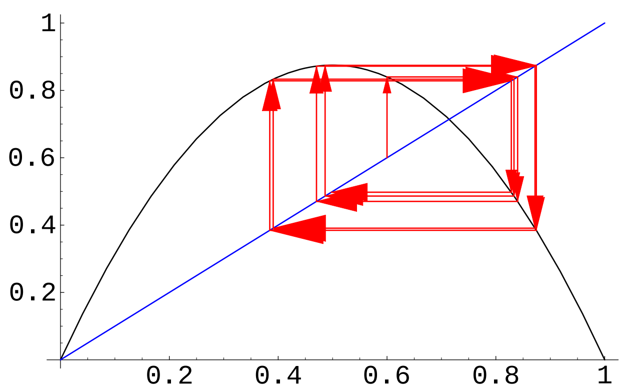










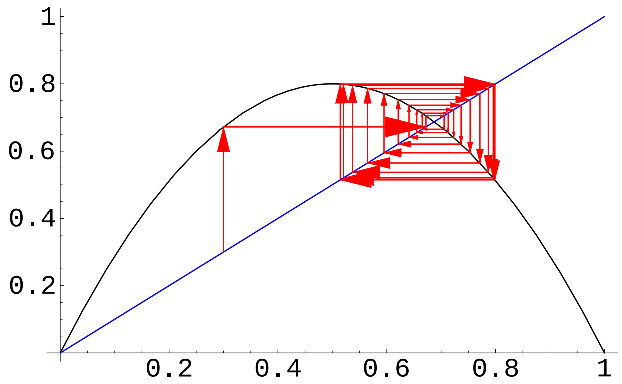



```
Out[31]= {- Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -, - Graphics -}
```

■ Two−cycles

So here at μ=3, we have a positive attracting fixed point, but the convergence is *very* slow. At this point $L_3{}'(p_\mu) = -1$, so we see *a* bifurcation.

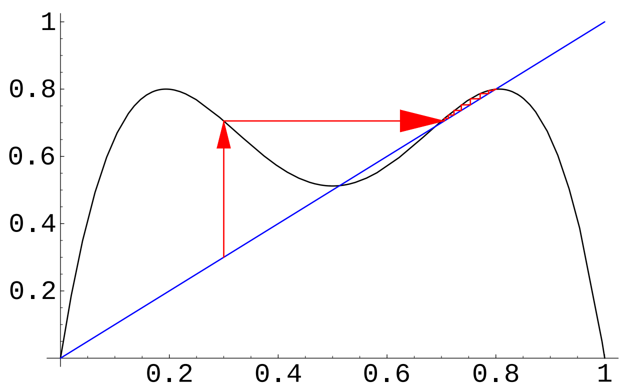
```
In[35]:= Plot[{L3.2[x], x}, {x, 0, 1},
          PlotStyle -> {{}, Blue},
          Epilog -> {Red,
             BouncePath[L3.2, 0.3, 20,
             HeadScaling -> Relative]}]
```



```
Out[35]= - Graphics -
```

It’s easier to see on the graph of $L_3{}^2$:

```
In[41]:= Plot[{L[3.2, L[3.2, x]], x}, {x, 0, 1},
         PlotStyle -> {{}, Blue},
          Epilog -> {Red,
             BouncePath[Nest[L3.2, #, 2]&, 0.3, 20,
             HeadScaling -> Relative]}]
```



```
Out[41]= - Graphics -
```

We can solve for these two−cycles explicitly:

```
In[14]:= Solve[L[μ, L[μ, x]] == x, x]
```

Out[14]= $\left\{\{x \to 0\}, \left\{x \to \frac{-1+\mu}{\mu}\right\}, \left\{x \to \frac{-(-1-\mu)\,\mu - \mu\sqrt{-3-2\,\mu+\mu^2}}{2\,\mu^2}\right\}, \left\{x \to \frac{-(-1-\mu)\,\mu + \mu\sqrt{-3-2\,\mu+\mu^2}}{2\,\mu^2}\right\}\right\}$

```
In[15]:= Simplify[%]
```

Out[15]= $\left\{\{x \to 0\}, \left\{x \to \frac{-1+\mu}{\mu}\right\}, \left\{x \to \frac{1+\mu-\sqrt{-3-2\,\mu+\mu^2}}{2\,\mu}\right\}, \left\{x \to \frac{1+\mu+\sqrt{-3-2\,\mu+\mu^2}}{2\,\mu}\right\}\right\}$

The first two solutions correspond to the "multiple roots" 0 and p_μ . The other two are the nontrivial ones, that is, the four − cycles .

```
In[16]:= (D[L[μ, L[μ, x]], x] /. %) // Simplify
```

Out[16]= $\{\mu^2, (-2+\mu)^2, 4+2\mu-\mu^2, 4+2\mu-\mu^2\}$

When are these two−cycles stable? When $\frac{d^2}{dx} L_\mu{}^2$ is between 1 and −1 at these points.

```
In[17]:= qpm = %[[3]]
         Solve[qpm == 1, μ]
         Solve[qpm == -1, μ]
```

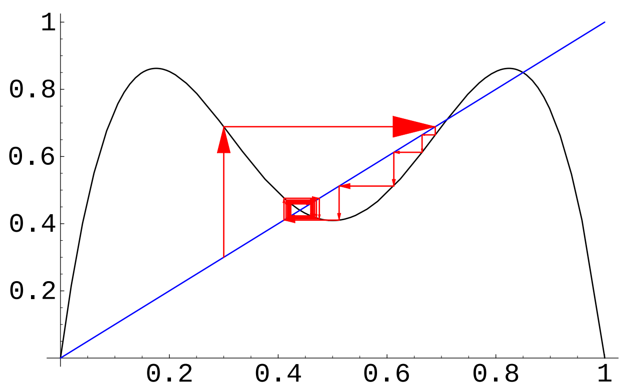
Out[17]= $4+2\mu-\mu^2$

Out[18]= $\{\{\mu \to -1\}, \{\mu \to 3\}\}$

Out[19]= $\{\{\mu \to 1-\sqrt{6}\}, \{\mu \to 1+\sqrt{6}\}\}$

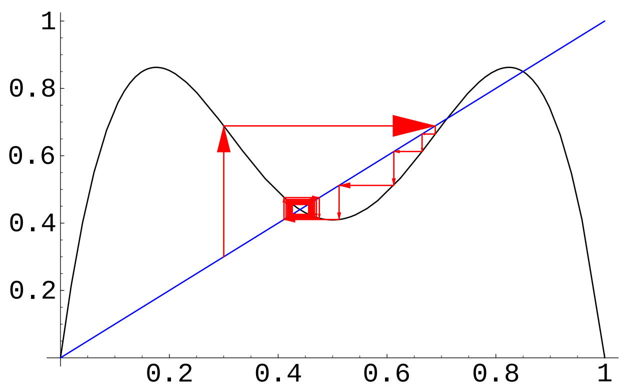
So we see the two−cycles are stable as long as $3 < \mu < 1 + \sqrt{6}$.

```
In[20]:= Show[
          Plot[L[1 + Sqrt[6], L[1 + Sqrt[6], x]], {x, 0, 1},
           DisplayFunction -> Identity],
          Graphics[{
            Blue,
            Line[{{0, 0}, {1, 1}}],
            Red,
            BouncePath[(L[1 + Sqrt[6], L[1 + Sqrt[6], #]]&), 0.3, 20,
             HeadScaling -> Relative]}],
         PlotRange -> All,
         DisplayFunction -> $DisplayFunction]
```



Out[20]= - Graphics -

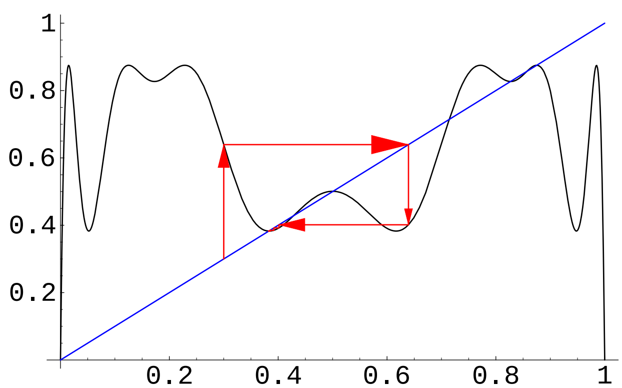
```
In[21]:= Plot[{Nest[L_{1+√6}, x, 2], x}, {x, 0, 1},
          PlotStyle -> {{}, Blue},
          Epilog -> {Red,
             BouncePath[Nest[L_{1+√6}, #, 2]&, 0.3, 30,
             HeadScaling -> Relative]}]
```



```
Out[21]= - Graphics -
```

After this critically stable 2−cycle, we expect to see another bifurcation, and in fact we do:

```
In[22]:= N[1. + Sqrt[6]]

Out[22]= 3.44949

In[23]:= L[n_Integer, μ_, x_] :=
          Nest[(L[μ, #]&), x, n]

In[24]:= L_{μ_,n_}[x_] := L[n, μ, x]

In[25]:= Plot[{L_{3.5,4}[x], x}, {x, 0, 1},
          PlotStyle -> {{}, Blue},
          Epilog -> {Red,
             BouncePath[L_{3.5,4}, 0.3, 20,
             HeadScaling -> Relative]}]
```

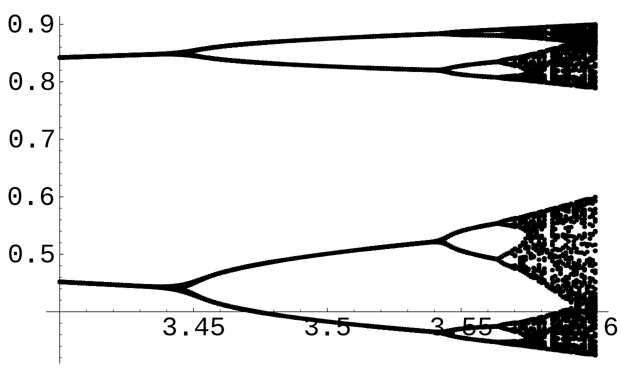


```
Out[25]= - Graphics -
```

■ Orbit Structure, Numeric

So we seem to be witnessing a period−doubling cascade as μ continues to increase. However, for $\mu = 4$, we already know the behavior, and that is chaotic as chaotic can be. To see what happens in between, we call on the computer again. This time, we ask for the limiting behavior of the system after a large number of iterates for different values of μ.

```
In[45]:= ω[μ_, x_, n_Integer, M_Integer] :=
          Union[NestList[Lμ, Nest[Lμ, x, n], M]]

In[46]:= ω[3.5, 0.5, 100, 20]

Out[46]= {0.38282, 0.500884, 0.826941, 0.874997}

In[56]:= pts = Flatten[
           Table[
          Map[{μ, #}&,
           ω[μ, 0.5, 100, 100]],
          {μ, 3.4, 3.6, 0.001}],
           1];

In[59]:= ListPlot[pts]
```

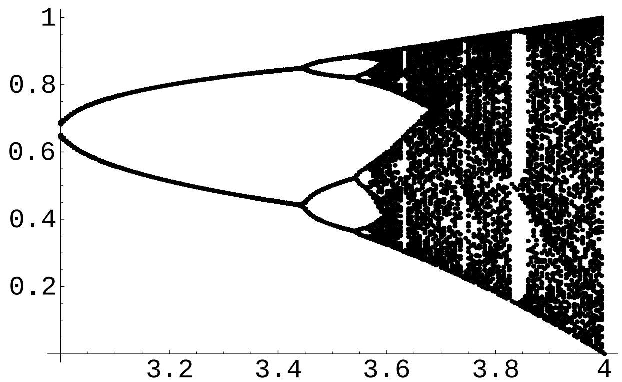



```
Out[59]= - Graphics -
```

So we see a very rapid period−doubling cascade, and it’s hard to tell what comes then. Let’s plot the full range (these are the slides from my lecture):

```
In[49]:= pts = Flatten[
           Table[
          Map[{μ, #}&,
           ω[μ, 0.5, 100, 100]],
          {μ, 3, 4, 0.005}],
           1];
```

```
In[50]:= ListPlot[pts]
```

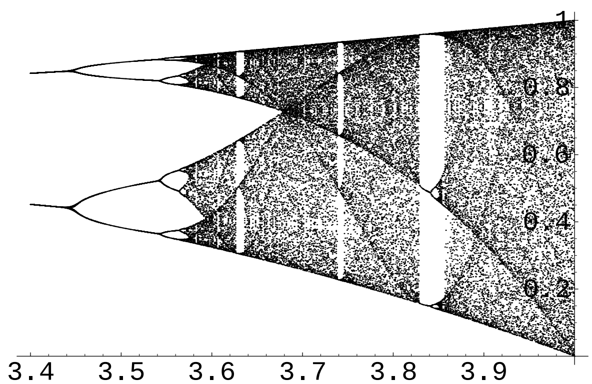



```
Out[50]= - Graphics -
```

Wow!

```
In[51]:= pts2 = Flatten[
           Table[
          Map[{μ, #}&,
           ω[μ, 0.5, 100, 100]],
          {μ, 3.4, 4, 0.001}],
           1];
```

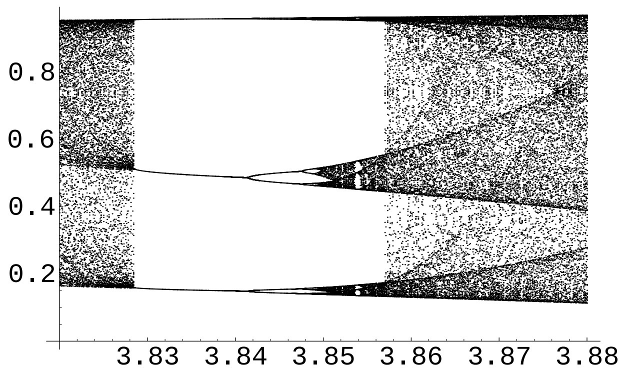
```
In[52]:= ListPlot[pts2,
         PlotStyle -> PointSize[0.002]]
```




```
Out[52]= - Graphics -
```

```
In[53]:= pts3 = Flatten[
           Table[
          Map[{μ, #}&,
           ω[μ, 0.5, 300, 100]],
          {μ, 3.82, 3.88, 0.0001}],
           1];
```

```
In[54]:= ListPlot[pts3,
         PlotStyle -> PointSize[0.002]]
```




```
Out[54]= - Graphics -
```

■ Scaling

■ Scaling in the period−doubling cascade

Look closely at the orbit diagram, especially the various period−doubling cascades. The bifurcation points all appear to be progressing geometrically. That is, the increase in parameter value needed to get the *next* bifurcation seems to be a constant times the total increas in parameter value need to get the current bifurcation. In words, if Λ_n is the point of the bifurcation to 2^n−cycles, then it appears that

$$\Lambda_{n+1} - \Lambda_n \sim \tilde{\delta}^{-n}$$

where δ is some constant. How can we find δ? Well, if we shift the above relation by one in the index, and divide, and then take the limit, we get

$$\tilde{\delta} = \lim_{n \to \infty} \frac{\Lambda_n - \Lambda_{n-1}}{\Lambda_{n+1} - \Lambda_n}$$

Now finding these numbers Λ_n is sort of tricky. We will have to solve the pair of equations

$$L_{\Lambda_n}(x) = x;$$
$$L_{\Lambda_n}'(x) = 1.$$

That is, we have to find the least stable 2^n−cycle. But there’s an easier way if instead we find the *most* stable 2^n−cycle. That is,

$$L_{\mu_n}(x) = x;$$
$$L_{\mu_n}'(x) = 0.$$

If the limit $\tilde{\delta}$ exists, then the limit of successive quotients of differences of successive μ_n’s will also converge to a number δ. Morever, these numbers are much easier to find numerically. Why? There are 2^n points on a superstable 2^n−cycle, but notice that the derivative along the cycle must be 0. If those points are labeled $x_1, x_2, \ldots, x_n$, then

$$0 = L_{\mu_n}'(x_1) = L_{\mu_n}'(x_1)\, L_{\mu_n}'(x_2) \ldots L_{\mu_n}'(x_{2^n}).$$

Since L_μ has a unique critical point for all μ, this means that one of the numbers x_1, x_2, ..., x_n must be that critical point, in this case, 1/2. Thus to find μ_n, we need only solve one equation:

$$L_{\mu_n}\left(\frac{1}{2}\right) = \frac{1}{2};$$

This is not hard to do. In fact, Newton's method lends itself nicely. Then we can find the successive approximations

$$\delta_n = \frac{\mu_n - \mu_{n-1}}{\mu_{n+1} - \mu_n}$$

and compute the limit.

Since we're going to be doing lots of iterations, let's compile the function so that we can game some speed and cut down on memory use.

```
In[60]:= LNc = Compile[
           {μ, x, {N, _Integer}},
           Module[{t},
            t = x; Do[t = μ t (1 - t), {N}]; t]]

Out[60]= CompiledFunction[{μ, x, N}, Module[{t}, t = x; Do[t = μ t (1 - t), {N}]; t],
          -CompiledCode-]

In[61]:= Timing[LNc[3.6, 0.5, 1000]]
         Timing[L[1000, 3.6, 0.5]]

Out[61]= {0. Second, 0.419253}

Out[62]= {0.04 Second, 0.419253}
```

Notice the big difference after many iterations!

```
In[63]:= dLNdμ [ μ_, x_, N_Integer, ϵ_: (10.)^(-10)] :=
          (LNc[ μ + ϵ, x, N] - LNc[μ - ϵ, x, N]) / (2 ϵ)
```

$$\frac{\text{LNc}[\mu+\epsilon, x, N] - \text{LNc}[\mu-\epsilon, x, N]}{2\,\epsilon}$$

```
In[64]:= Table[Factor[L_{μ,2^n}[1/2] - 1/2], {n, 0, 2}]
```

Out[64]= $\left\{\frac{1}{4}(-2+\mu),\ -\frac{1}{16}(-2+\mu)(-4-2\mu+\mu^2),\ -\frac{1}{65536}((-2+\mu)(-4-2\mu+\mu^2)(4096+2048\mu^2-512\mu^3-1024\mu^4+64\mu^6+384\mu^7-192\mu^8-40\mu^9+48\mu^{10}-12\mu^{11}+\mu^{12}))\right\}$

So in finding our first couple of superstable parameter values, we have $\mu_0 = 2$.

In[65]:= $L_{\mu,2}$[x]

Out[65]= $(1-x)\,x\,\mu^2\,(1-(1-x)\,x\,\mu)$

```
In[66]:= Solve[(% /. {x -> 1/2}) == 1/2, μ]
```

Out[66]= $\{\{\mu \to 2\}, \{\mu \to 1-\sqrt{5}\}, \{\mu \to 1+\sqrt{5}\}\}$

```
In[67]:= N[%]

Out[67]= {{μ → 2.}, {μ → -1.23607}, {μ → 3.23607}}
```

And $\mu_1 = 1 + \sqrt{5} = 3.23607....$ The problem is that as n grows, the polynomial we are trying to find the root of grows exponentially in degree. In fact, $L^N{}_\mu$ has degree 2^N. So not only is symbolic solving (which involves factoring) going to take way too much time, we're going to have too many choices for roots. So we need to be more clever in solving.

We use Newton's method, but give an initial guess close to where we want it.

```
In[68]:= FindSS[NN_, μ_: 3, maxSteps_: 1000] :=
  FixedPoint[(# - (LNc[#, 0.5, NN] - 0.5)/(dLNdμ[#, 0.5, NN])) &,
   μ, maxSteps]

        General::spell1 :
         Possible spelling error: new symbol name "maxSteps" is similar to existing symbol "MaxSteps".
```

Looking at the orbit diagram shows that μ_2 is close to 3.45, so we try that.

```
In[69]:= FindSS[4, 3.45]

Out[69]= 3.49856
```

So $\mu_2 = 3.49856...$.

```
In[70]:= μss = {2, 1 + Sqrt[5], %}

Out[70]= {2, 1 + √5, 3.49856}
```

In order to get better first guesses, we use the assumption that the quotients δ_n form a convergent sequence. That is, we can rewrite

$$\mu_{n+1} = \mu_n + \frac{\mu_n - \mu_{n-1}}{\delta_n}$$

But, since the δ_n are converging, we can *approximate*

$$\mu_{n+1} \approx \mu_n + \frac{\mu_n - \mu_{n-1}}{\delta_{n-1}}$$

and hope that we are close enough to μ_{n+1} to converge to it under iterations of Newton's method.

Let's code this all up into some functions on lists.

```
In[71]:= δ[list_List?VectorQ,
   n_Integer: (-1)] := (list[[n - 1]] - list[[n - 2]])/(list[[n]] - list[[n - 1]])

        General::spell1 :
         Possible spelling error: new symbol name "list" is similar to existing symbol "List".
```

Note the use of the optional argument here. Without the last argument, we assume the value −1, which, in terms of list elements, is the *last* one. Here is our first approximation to δ:

```
In[72]:= δ[μss]

Out[72]= 4.70894
```

Here we also code a **`NextGuess`** function that gives a guess for μ_{n+1}.

```
In[73]:= NextGuess[list_List: VectorQ,
         n_: (-1)] :=
         list[[n]] + (list[[n]] - list[[n - 1]]) / δ[list, n]

In[74]:= NextGuess[μss]

Out[74]= 3.55431

In[75]:= FindSS[8, %]

Out[75]= 3.55464
```

Just like that, we have found μ_3.

Let's loop this and find a whole bunch of μ_n at once. Note the use of the **`Print`** statement to assuage the impatient. At least we get a progress report. ☺

```
In[76]:= mylist = Module[
          {μlist = μss, NN, i, newμ, time},
          NN = 2^(Length[μlist] - 1);
          For[i = 3, i < 13, i++,
             NN *= 2;
             {time, newμ} = Timing[FindSS[NN, NextGuess[μlist]]];
             AppendTo[μlist, newμ];
             Print[i, "\t",
              N[newμ, 20], "\t",
              N[δ[μlist], 20], "\t",
             time]];
           μlist]

        3 3.554640862768825 4.680770998010699 0. Second

        4 3.566667379856268 4.66295961111428 0. Second

        5 3.569243531637111 4.668403925916237 0. Second

        6 3.569795293749947 4.668953740957641 0.01 Second

        7 3.569913465422345 4.669157181568272 0. Second

        8 3.569938774233306 4.669191001397932 0. Second

        9 3.569944194608065 4.669199472334519 2.32 Second

        10 3.569945355486475 4.669201106961582 4.41 Second

        11 3.569945604111073 4.669201762692549 8.67 Second

        12 3.569945657358845 4.66920189526357 0.22 Second

Out[76]= {2, 1 + √5, 3.49856, 3.55464, 3.56667, 3.56924, 3.5698, 3.56991, 3.56994,
          3.56994, 3.56995, 3.56995, 3.56995}

In[77]:= μ∞ = mylist[[-1]]

Out[77]= 3.56995
```

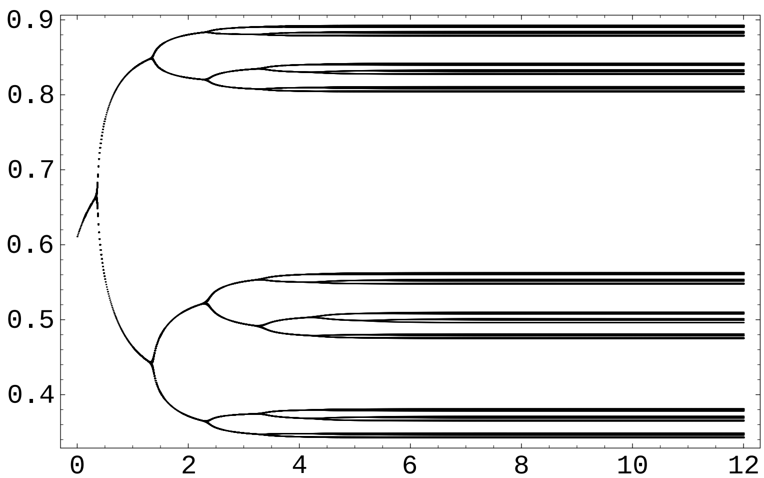
```
In[78]:= δ∞ = δ[mylist]

Out[78]= 4.6692
```

Now that we know this δ, we can plot the μ−axis in our orbit diagram exponentially. Perform the change of coordinates

$$\nu = -\frac{\log(\mu_\infty - \mu)}{\log \delta\,.}$$

Then we plot anew:

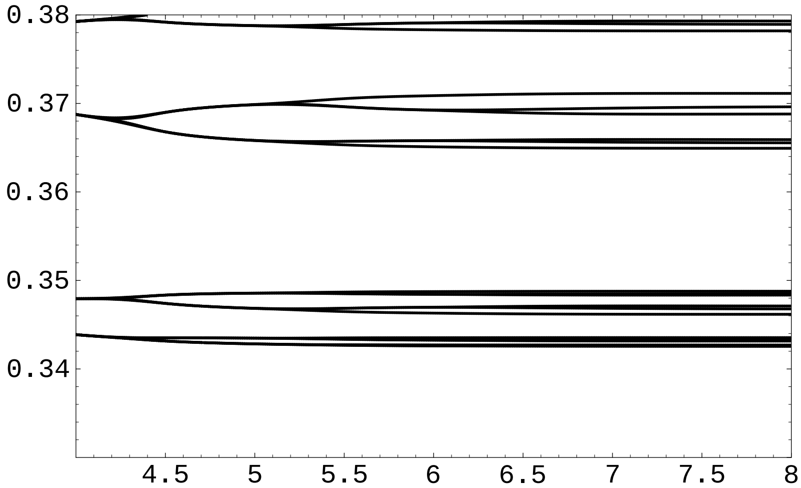
```
In[79]:= pts5 = Flatten[
            Table[
           Map[{ν, #}&,
            ω[μ∞ - Exp[-ν Log[δ∞]], 0.5, 100, 100]],
          {ν, 0, 12, 0.01}],
            1];

In[80]:= ListPlot[pts5,
         Axes -> False, Frame -> True,
         PlotStyle -> PointSize[0.002]]
```




```
Out[80]= - Graphics -
```

As predicted, in these coordinates, we see bifurcation happening at regular intervals.

```
In[81]:= ListPlot[pts5,
         Axes -> False, Frame -> True,
         PlotStyle -> PointSize[0.004],
         PlotRange -> {{4, 8}, {0.33, 0.38}}]
```




```
Out[81]= - Graphics -
```

■ Scaling in state

So we have found the scaling factor δ in "parameter space" parameterized by μ. But there is also a scaling factor in *state space*, i.e., the vertical axis of the orbit diagram. Not only is the length of each bifurcation decreasing geometrically, so is the width of each fork in the tree. To find this scaling factor α, we define d_n to be the distance between 1/2 and the next closest point on the superstable 2^n−cycle. What is that point? Well, before the nth bifurcation, those two points were actually the *same* point on a 2^{n-1}−cycle. The point closest to 1/2 is the point that is halfway done to coming back to 1/2. That is,

$$d_n = \boldsymbol{L}_{\mu_n}{}^{2^{n-1}}\left(\frac{1}{2}\right) - \frac{1}{2}$$

Then we are seeking

$$\alpha = \lim_{n\to\infty}\frac{d_{n+1}}{d_n} = \lim_{n\to\infty}\frac{\boldsymbol{L}_{\mu_{n+1}}{}^{2^n}\left(\frac{1}{2}\right) - \frac{1}{2}}{\boldsymbol{L}_{\mu_n}{}^{2^{n-1}}\left(\frac{1}{2}\right) - \frac{1}{2}}$$

Since we already have the list of superstable cycle parameter values, we can do this pretty easily with a simple incrementer. Study the code below until you understand how it works. The main line is of course the second.

```
In[82]:= n = 0;
         dlist = Map[
           (LNc[#, 0.5, 2^(n++)] - 1 / 2)&,
          Rest[mylist]]
         n =.

        General::spell1 :
         Possible spelling error: new symbol name "dlist" is similar to existing symbol "list".

Out[82]= {0.309017, -0.116402, 0.0459752, -0.0183262, 0.00731843, -0.00292368,
          0.00116809, -0.00046669, 0.000186459, -0.0000744969, 0.0000297641, -0.0000118918}
```

Now we just take successive quotients.

```
In[84]:= N[Drop[dlist, -1] / Drop[dlist, 1], 20]

Out[84]= {-2.654744816835617, -2.531837659337503, -2.508718199034679, -2.504112789194416,
          -2.503161125691425, -2.502961188518201, -2.502919174358635, -2.502910275261882,
          -2.50290836414096, -2.502908049703835, -2.502908603397627}

In[85]:= α∞ = Last[%]

Out[85]= -2.502908603397627
```

We have thus found an approximation to α.

■ Universality

Notice that the experiments we carried out had very little to do with L_μ being a parabola. All that is really necessary is that it be "parabola−like." We can repeat the experiment with any family of functions that map a fixed interval to itself and are *unimodal*; i.e., they have a single quadratic maximum.

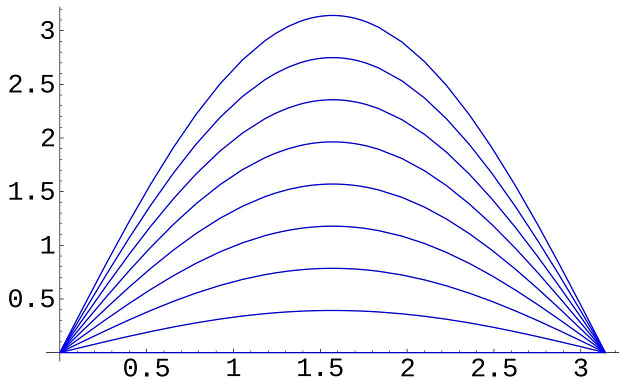
■ A sine family

Consider the function

```
In[88]:= S[λ_, x_] := λ Sin[x]

In[89]:= Sλ_[x_] := S[λ, x]
```

as λ ranges over the interval $[0 2\pi]$.

```
In[90]:= Plot[Evaluate[Table[S[λ, x], {λ, 0, Pi, π / 8}]],
          {x, 0, Pi},
         PlotStyle -> {RGBColor[0, 0, 1]}]
```



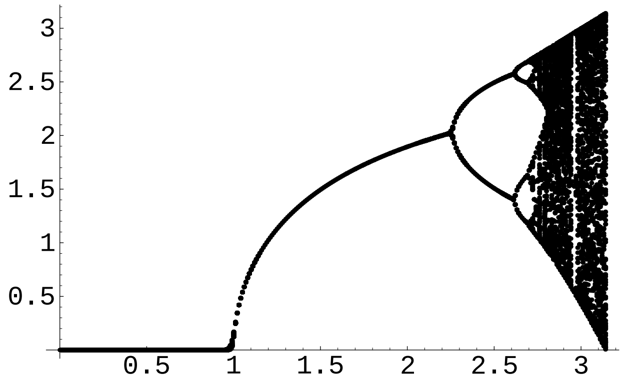

```
Out[90]= - Graphics -
```

Again, we will be able to find fixed points and cycles, but this will be infinitely harder because of the transcendental nature of sine. However, we can still plot the orbit diagram.

```
In[91]:= ωSin[λ_, x_, n_Integer, M_Integer] :=
          Union[NestList[Sλ, Nest[Sλ, x, n], M]]
```

```
In[92]:= pts = Flatten[
            Table[
          Map[{λ, #}&,
           ωSin[λ, N[Pi / 2], 100, 100]],
          {λ, 0.0, Pi, 0.01}],
           1];
```

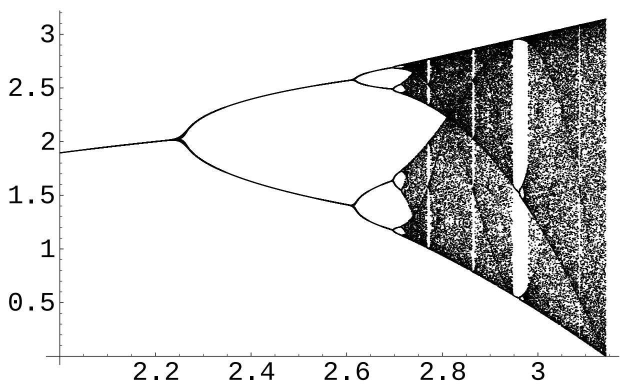
```
In[93]:= ListPlot[pts]
```




```
Out[93]= - Graphics -
```

```
In[94]:= pts2 = Flatten[
            Table[
          Map[{λ, #}&,
           ωSin[λ, N[Pi / 2], 100, 100]],
          {λ, 2.0, Pi, 0.001}],
           1];
```

```
In[95]:= ListPlot[pts2,
         PlotStyle -> PointSize[0.002]]
```




```
Out[95]= - Graphics -
```

As boasted, we see a period−doubling cascade with exactly the same behavior. Note even the appearance of the band of stable three−cycles!

Let's try to find δ and α for this family. The first three values of δ_n we can solve using some of *Mathematica*'s built−in functions and some guesswork.

```
In[96]:= pi2 = N[Pi / 2]
```

```
Out[96]= 1.5708
```

```
In[97]:= NSolve[S[λ, pi2] == pi2, λ]

Out[97]= {{λ → 1.5708}}

In[98]:= FindRoot[S[λ, S[λ, pi2]] == pi2, {λ, 2.1}]

Out[98]= {λ → 2.44332}

In[99]:= FindRoot[Nest[S[λ, #]&, pi2, 4] == pi2, {λ, 2.7}]

Out[99]= {λ → 2.65899}

In[100]:= δ[{pi2, 2.44332, 2.65899}]

Out[100]= 4.04564

In[101]:= SNc = Compile[
            {λ, x, {N, _Integer}},
            Module[{t},
             t = x; Do[t = λ Sin[t], {N}]; t]]

Out[101]= CompiledFunction[{λ, x, N}, Module[{t}, t = x; Do[t = λ Sin[t], {N}]; t], -CompiledCode-]
```

In[102]:= dSNdλ [λ_, x_, N_Integer, ϵ_: (10.) ^ (-10)] :=

$$\frac{\text{SNc}[\lambda + \epsilon, x, N] - \text{SNc}[\lambda - \epsilon, x, N]}{2\,\epsilon}$$

In[103]:= FindSSSin[NN_, λ_: 3, maxSteps_: 1000] :=

$$\text{FixedPoint}\left[\left(\# - \frac{\text{SNc}[\#, \text{pi2}, \text{NN}] - \text{pi2}}{\text{dSNd}\lambda[\#, \text{pi2}, \text{NN}]}\right)\&,\ \lambda,\ \text{maxSteps}\right]$$

```
In[104]:= FindSSSin[1, 1.5]

Out[104]= 1.5708

In[105]:= λss = Apply[FindSSSin, {{1, 1.5}, {2, 2.2}, {4, 2.6}}, {1}]

        General::spell1 :
         Possible spelling error: new symbol name "λss" is similar to existing symbol "μss".

Out[105]= {1.5708, 2.44332, 2.65899}
```

In[106]:=
```
mylist = Module[
  {λlist = λss, NN, i, newλ, time},
  NN = 2^(Length[λlist] - 1);
  For[i = 3, i < 13, i++,
     NN *= 2;
     {time, newλ} = Timing[FindSSSin[NN, NextGuess[λlist]]];
     AppendTo[λlist, newλ];
     Print[i, "\t",
      N[newλ, 20], "\t",
      N[δ[λlist], 20], "\t",
     time]];
   λlist]
```

```
General::spell1 :
 Possible spelling error: new symbol name "newλ" is similar to existing symbol "newμ".

General::spell1 :
 Possible spelling error: new symbol name "λlist" is similar to existing symbol "μlist".

3 2.706326093765227 4.555852749921388 0.01 Second

4 2.716516885833067 4.645181717674329 0. Second

5 2.718701840461667 4.664074912333342 0. Second

6 2.719169900694699 4.668105671883451 0. Second

7 2.719270149906629 4.668966708276939 0. Second

8 2.719291620449081 4.669151333847085 0.01 Second

9 2.719296218792398 4.669190830968664 2.06 Second

10 2.719297203617199 4.669199344778925 3.83 Second

11 2.719297414536538 4.669201062731633 0.06 Second

12 2.719297459708998 4.669202008947444 14.9 Second
```

Out[106]=
```
{1.5708, 2.44332, 2.65899, 2.70633, 2.71652, 2.7187, 2.71917, 2.71927,
 2.71929, 2.7193, 2.7193, 2.7193, 2.7193}
```

Voila! Notice that the value λ_∞ to which the superstable cycle points are converging is different from μ_∞ above. However, we do get the same δ.

In[107]:=
```
n = 0;
dlist = Map[
  (SNc[#, pi2, 2^(n++)] - pi2)&,
  Rest[mylist]]
n =.
```

Out[107]=
```
{0.872526, -0.336792, 0.133574, -0.0532873, 0.0212834, -0.00850289, 0.00339716,
 -0.00135728, 0.000542281, -0.00021666, 0.0000865635, -0.0000345851}
```

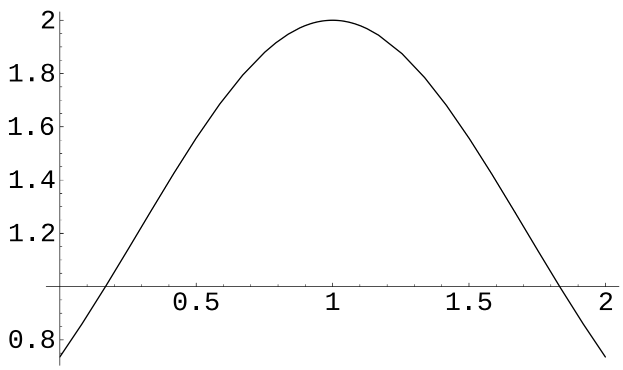
In[109]:=
```
N[Drop[dlist, -1] / Drop[dlist, 1], 20]
```

Out[109]=
```
{-2.590696934756435, -2.521384354804959, -2.506683425305186, -2.503702628614742,
 -2.503075349502518, -2.502943355720226, -2.502915409846389, -2.502909473307085,
 -2.50290824345891, -2.50290798684546, -2.502908516216082}
```

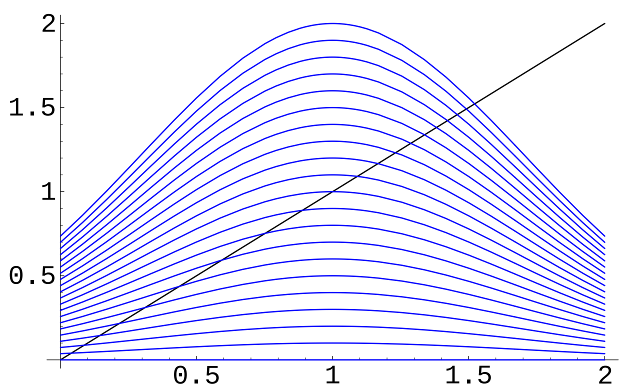
And the same α!

■ An exponential family

```
In[110]:= e[λ_, x_] := λ Exp[-(x - 1)^2]

In[111]:= e_λ_[x_] := e[λ, x]

In[112]:= Plot[e[2, x], {x, 0, 2}]
```




```
Out[112]= - Graphics -

In[113]:= Plot[Evaluate[Table[e[λ, x], {λ, 0, 2, 0.1}]],
           {x, 0, 2},
          PlotStyle -> {RGBColor[0, 0, 1]},
          Prolog -> {Line[{{0, 0}, {2, 2}}]}]
```

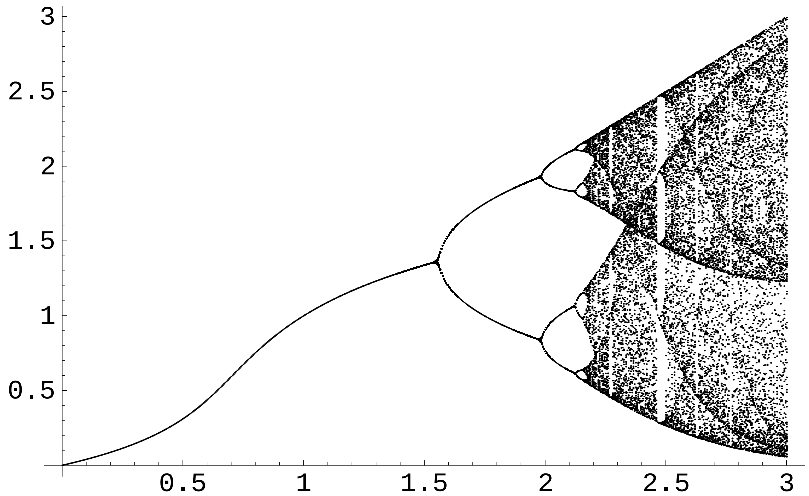



```
Out[113]= - Graphics -
```

Notice this family doesn't fix the endpoints of an interval. Nevertheless, we can still call it unimodal.

```
In[114]:= ωExp[λ_, x_, n_Integer, M_Integer] :=
           Union[NestList[e_λ, Nest[e_λ, x, n], M]]

In[115]:= pts = Flatten[
             Table[
           Map[{λ, #}&,
             ωExp[λ, N[1], 100, 100]],
           {λ, 0, 3, 0.005}],
             1];
```

```
In[116]:= ListPlot[pts, PlotRange -> All,
          PlotStyle -> {PointSize[0.002]}]
```

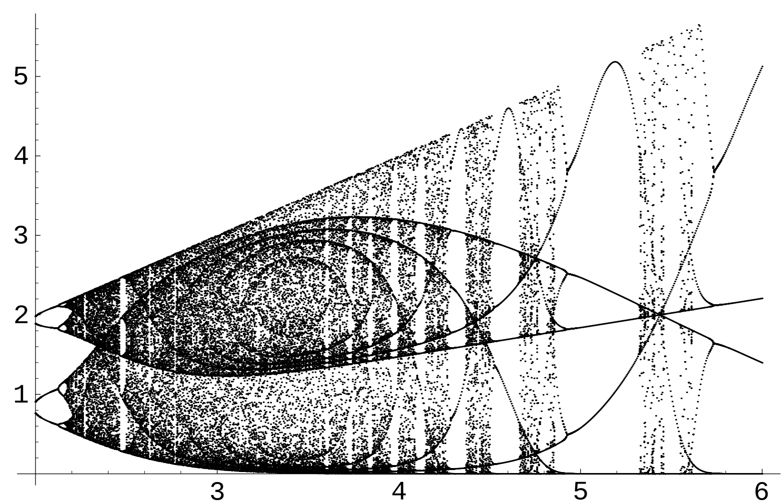



```
Out[116]= - Graphics -
```

What do we see? The curves are a little kinkier, and the region of chaos looks like it has "stripes" of density to it. But in the middle, we see the same, basic, period−doubling cascade. We can find the superstable cycles, and compute α and δ, and they will be the same.

Note, however, that we haven't said anything about what happens after the period−doubling cascade. That can also be analayzed, but all we're equipped to do is look at the pictures. For instance, this exponential orbit diagram looks quite different in the chaotic region:

```
In[117]:= pts = Flatten[
            Table[
           Map[{λ, #}&,
            ωExp[λ, N[1], 100, 100]],
           {λ, 2, 6, 0.005}],
            1];

In[118]:= ListPlot[pts, PlotRange -> All,
          PlotStyle -> {PointSize[0.002]}]
```




```
Out[118]= - Graphics -
```