Physics 3180: Computational Physics

Prof. Derek Jackson Kimball
Physics Dept.
CSU East Bay

Project 2: Chaos in the Logistic Map

Part A: x[n] vs. n

Function definition

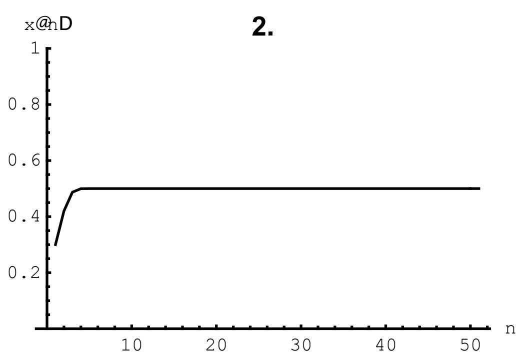
First we want to write a function that will calculate the behavior of the population for different values of the food parameter.

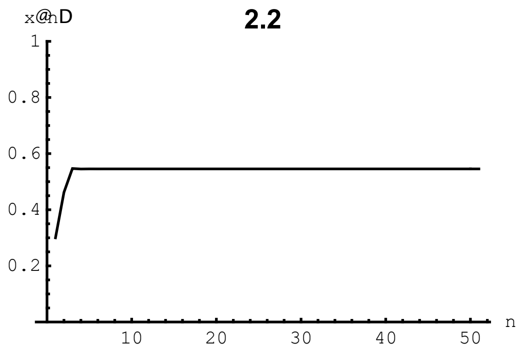
```
logisticFn[μ_, y_] := μ y (1 - y);
```

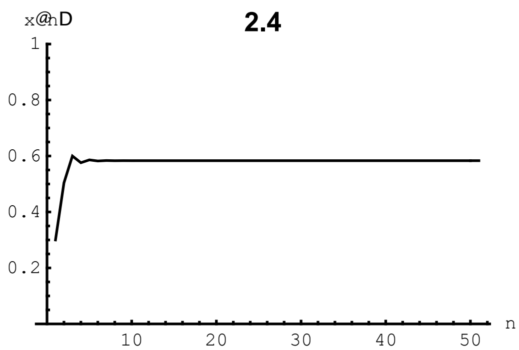
```
population[x0_, μ_, n_] := NestList[logisticFn[μ, #] &, x0, n]
```

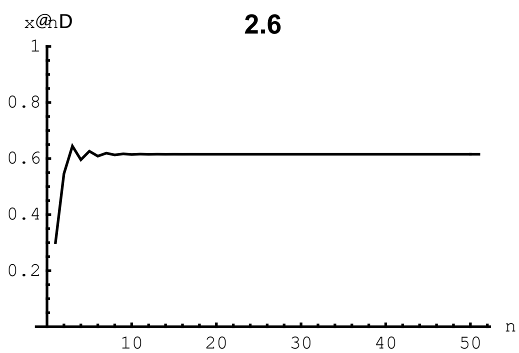
Plots

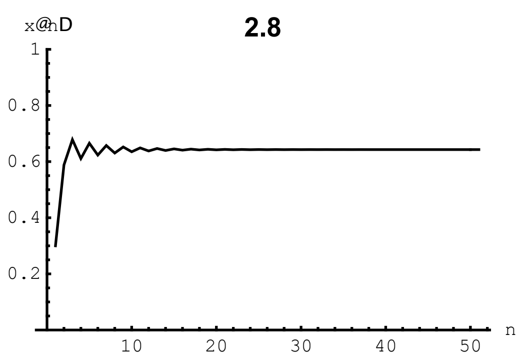
```
Table[
  ListPlot[population[0.3, 2.0 + j, 50], PlotRange → {0, 1}, PlotJoined → True,
   PlotLabel → StyleForm[2.0 + j, Section], AxesLabel → {"n", "x[n]"}], {j, 0, 2, .2}];
```

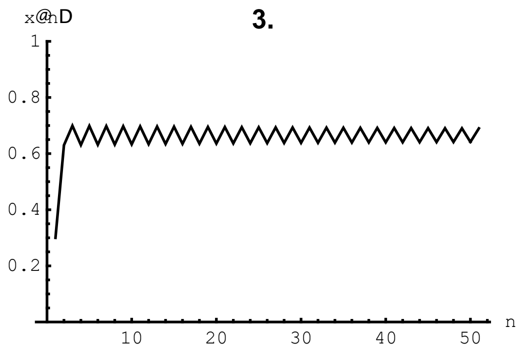


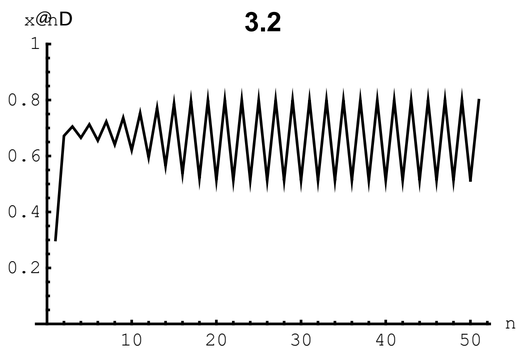

x@nD
2.2
1
0.8
0.6
0.4
0.2
10 20 30 40 50
n

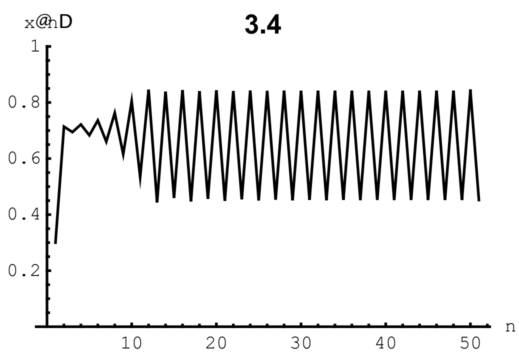

x@nD
2.4
1
0.8
0.6
0.4
0.2
10 20 30 40 50
n

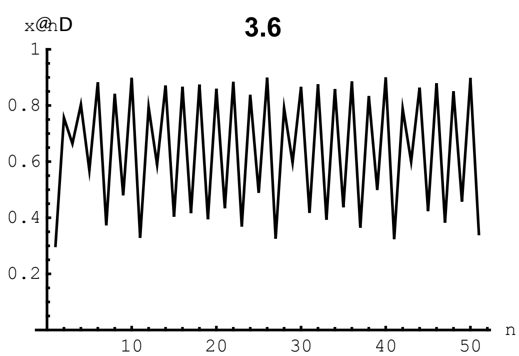

x@nD
2.6
1
0.8
0.6
0.4
0.2
10 20 30 40 50
n

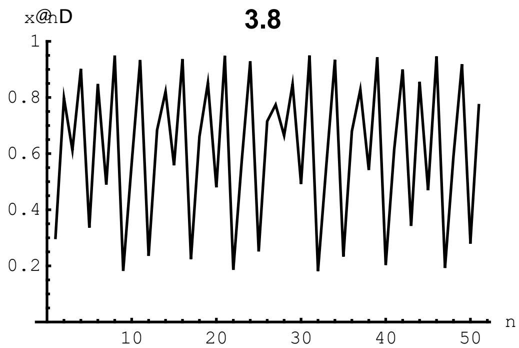

x@nD
2.8
1
0.8
0.6
0.4
0.2
10 20 30 40 50
n

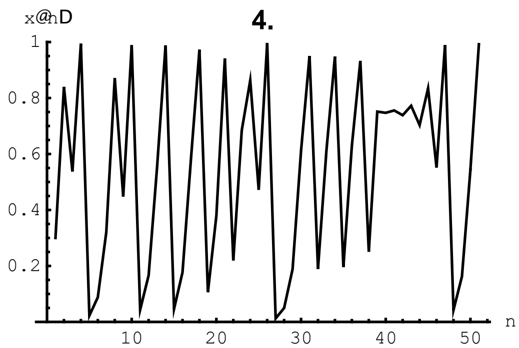

3.
x@nD
1
0.8
0.6
0.4
0.2
10
20
30
40
50
n


3.2
x@nD
1
0.8
0.6
0.4
0.2
10
20
30
40
50
n


3.4
x@nD
1
0.8
0.6
0.4
0.2
10
20
30
40
50
n


3.6
x@nD
1
0.8
0.6
0.4
0.2
10
20
30
40
50
n





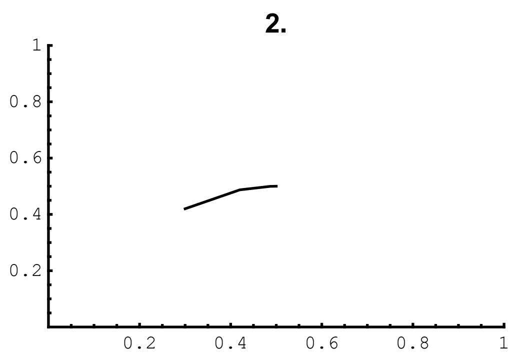
Part B: Iterative Plot, x[n+1] vs. x[n]

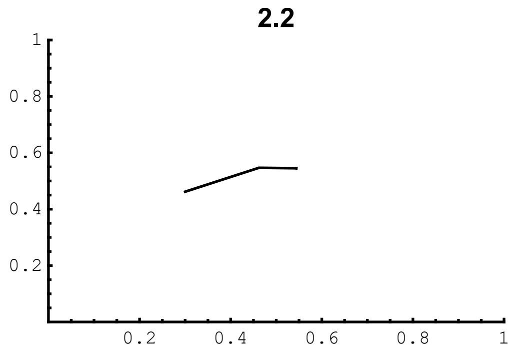
Function Definition

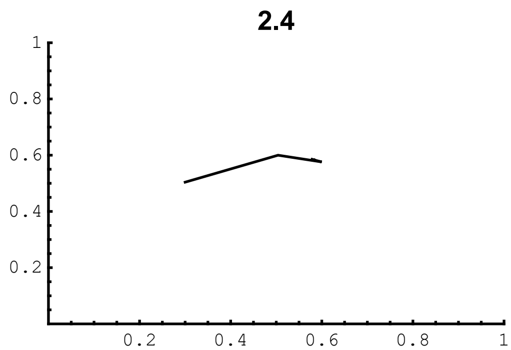
```
populationIterative[x0_, μ_, n_, order_] :=
 Most[(Transpose[{#, RotateLeft[#, order]}] &)[population[x0, μ, n]]]
```

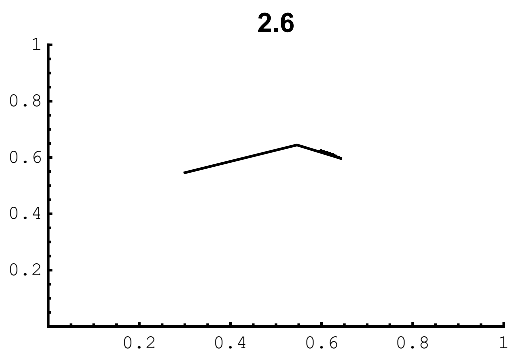
Plots

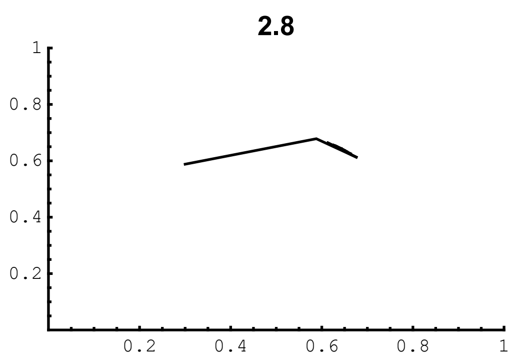
```
Table[
  ListPlot[populationIterative[0.3, 2.0 + j, 50, 1], PlotRange → {{0, 1}, {0, 1}},
   PlotJoined → True, PlotLabel → StyleForm[2.0 + j, Section]],
  {j, 0, 2, .2}];
```

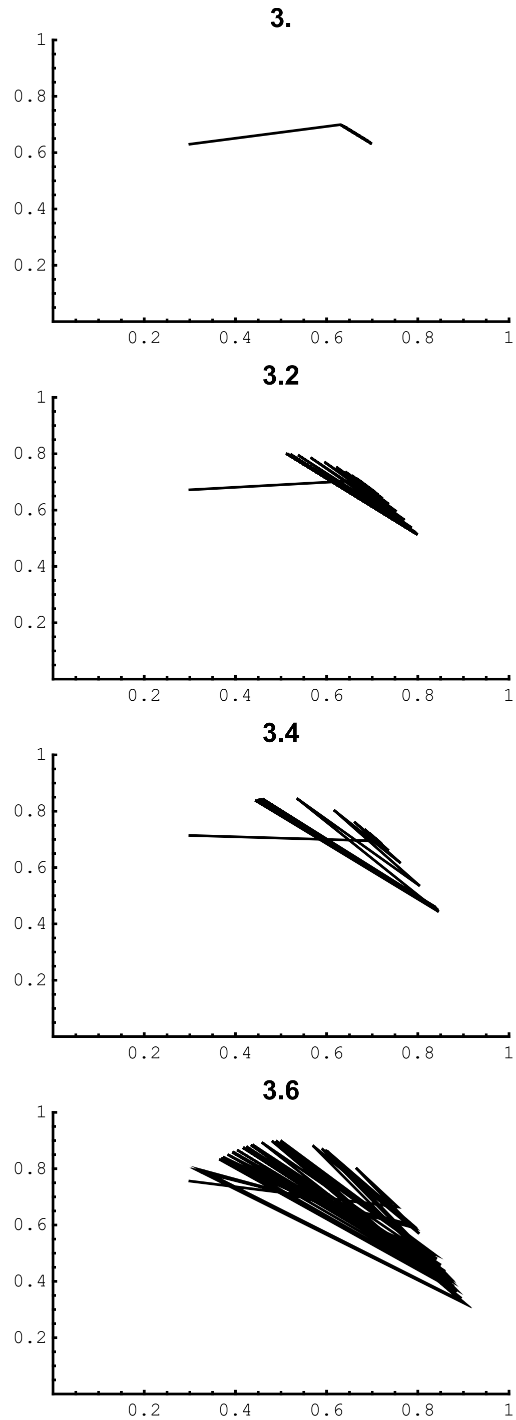


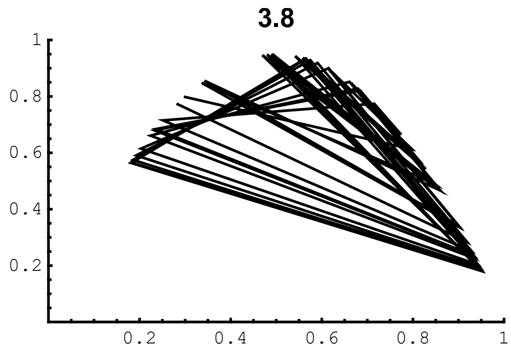

2.2
1
0.8
0.6
0.4
0.2
0.2 0.4 0.6 0.8 1

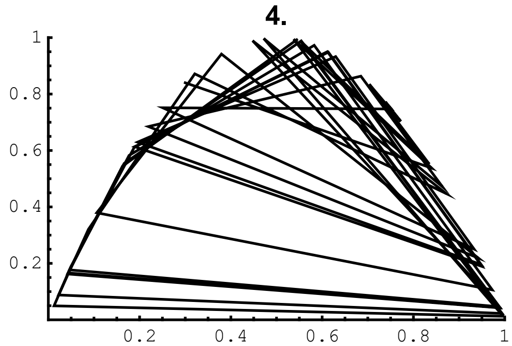

2.4
1
0.8
0.6
0.4
0.2
0.2 0.4 0.6 0.8 1


2.6
1
0.8
0.6
0.4
0.2
0.2 0.4 0.6 0.8 1


2.8
1
0.8
0.6
0.4
0.2
0.2 0.4 0.6 0.8 1


3.
3.2
3.4
3.6
0.2
0.4
0.6
0.8
1



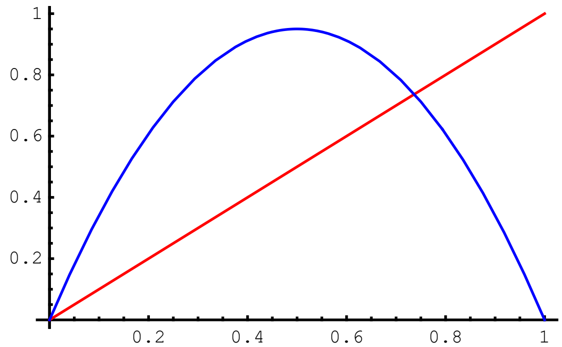


Part C: Fixed Points

Analysis: Fixed Points

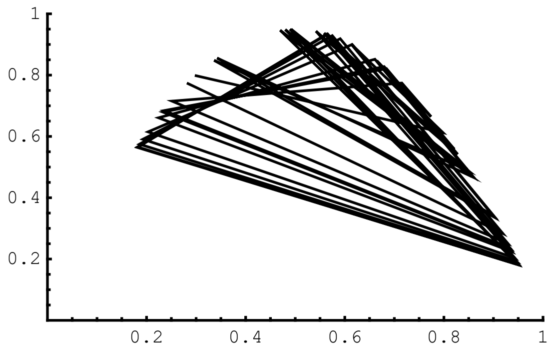
```
mu = 3.8;
```

```
logisticMapFixPointPlot1 = Plot[{y, logisticFn[mu, y]}, {y, 0, 1},
  PlotRange → All, PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}]
```



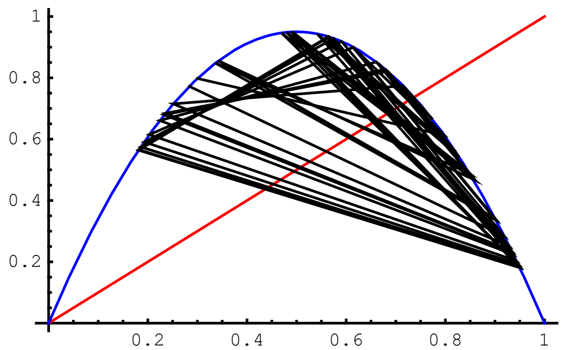

```
- Graphics -
```

```
logisticMapFixIterativePlot1 = ListPlot[
  populationIterative[0.3, mu, 50, 1], PlotRange → {{0, 1}, {0, 1}}, PlotJoined → True]
```



- Graphics -

```
Show[logisticMapFixPointPlot1, logisticMapFixIterativePlot1]
```

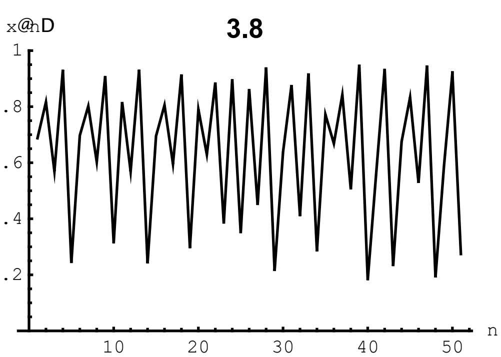


- Graphics -

```
Solve[y == logisticFn[mu, y], y]
```

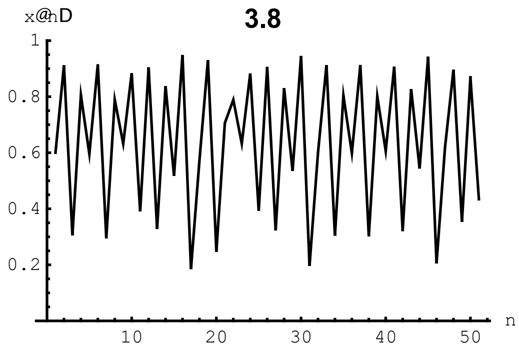
{{y → 0.}, {y → 0.736842}}

```
ListPlot[population[0.6875, mu, 50], PlotRange → {0, 1}, PlotJoined → True,
 PlotLabel → StyleForm[mu, Section], AxesLabel → {"n", "x[n]"}]
```



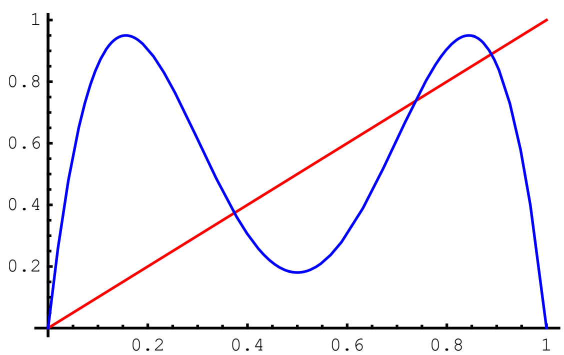
- Graphics -

```
ListPlot[population[0.6, mu, 50], PlotRange → {0, 1}, PlotJoined → True,
 PlotLabel → StyleForm[mu, Section], AxesLabel → {"n", "x[n]"}]
```



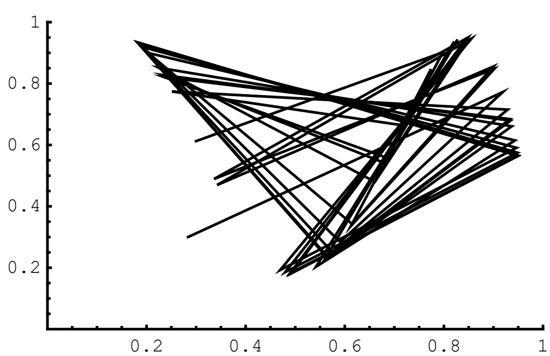
- Graphics -

```
logisticMapFixPointPlot2 = Plot[{y, logisticFn[mu, logisticFn[mu, y]]},
  {y, 0, 1}, PlotRange → All, PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}]
```



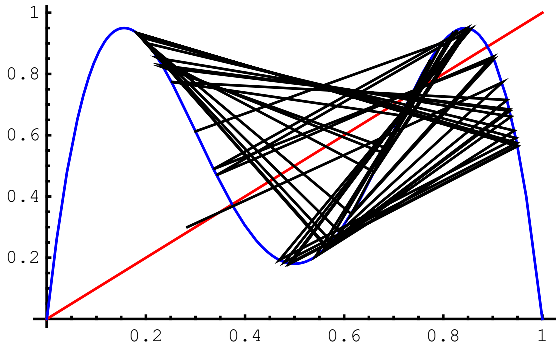
- Graphics -

```
logisticMapFixIterativePlot2 = ListPlot[
  populationIterative[0.3, mu, 50, 2], PlotRange → {{0, 1}, {0, 1}}, PlotJoined → True]
```



- Graphics -

```
Show[logisticMapFixPointPlot2, logisticMapFixIterativePlot2]
```




```
- Graphics -
```

```
Solve[y == logisticFn[mu, logisticFn[mu, y]], y]
```

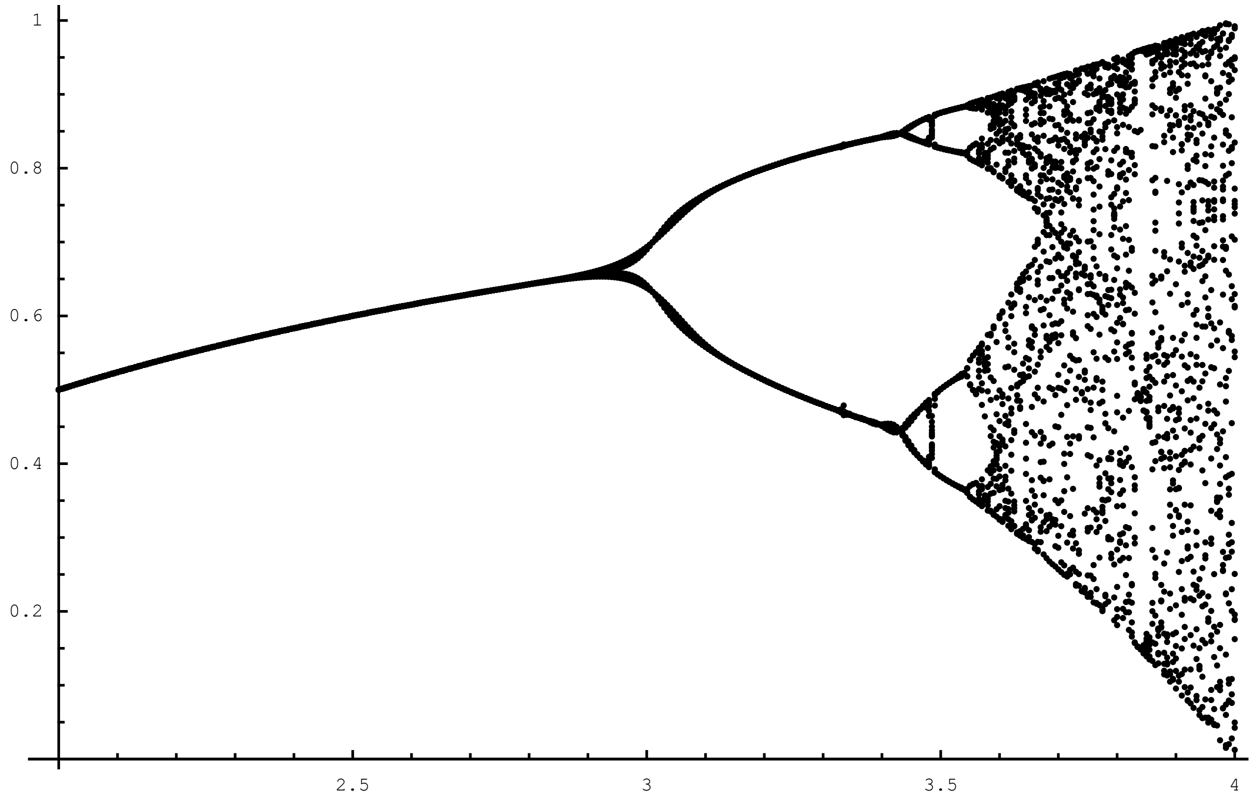
```
{{y → 0.}, {y → 0.373738}, {y → 0.736842}, {y → 0.88942}}
```

Part D: Bifurcation Diagram

```
populationMuList[x0_, μ_, n_, nInit_] := Take[{μ, #} & /@ population[x0, μ, n], -nInit]

populationBifurcation[x0_, n_, nInit_, μMin_, μMax_, steps_] :=
 Flatten[Table[populationMuList[x0, m, n, nInit], {m, μMin, μMax, (μMax - μMin) / steps}], 1]
```

```
ListPlot[populationBifurcation[0.3, 50, 25, 2.0, 4.0, 400],
 PlotRange → All, PlotJoined → False, PlotStyle → PointSize[.005]]
```



- Graphics -