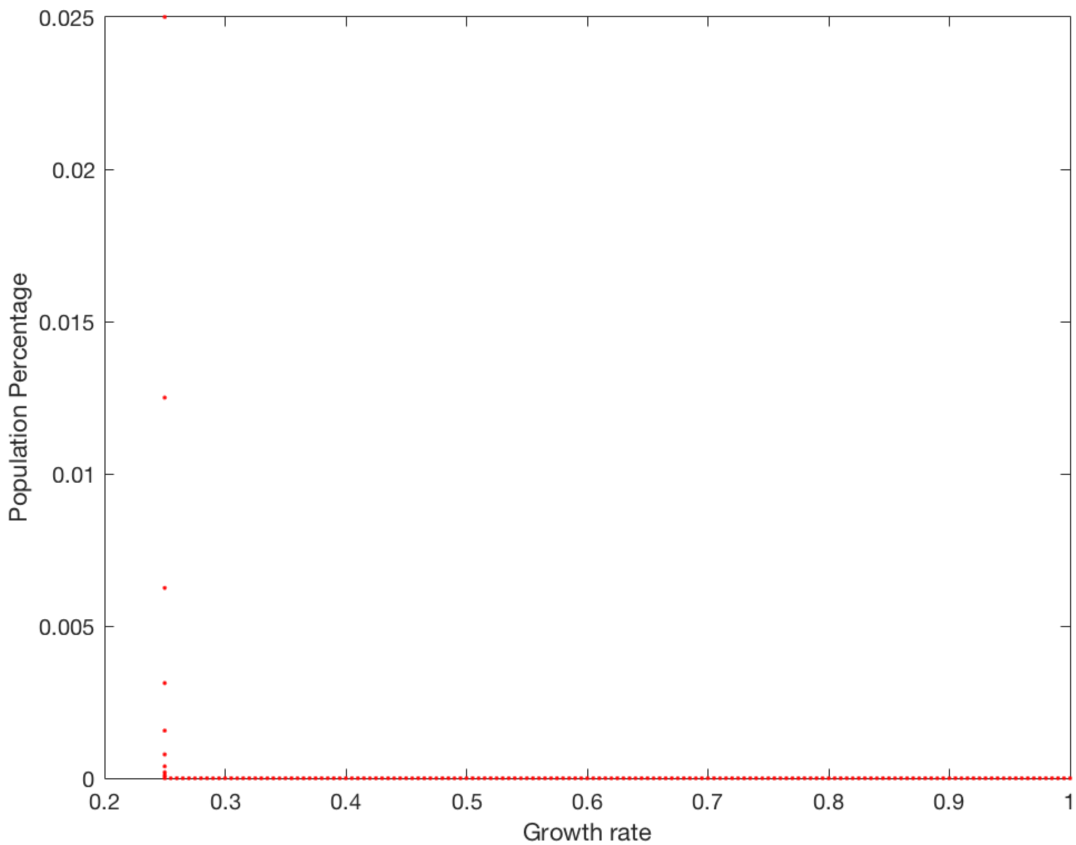
Rac158, R. Aidan Campbell
10-17-2016
EECS131 Quiz 4:

On Chaotic Functions and Bifurcation Diagrams

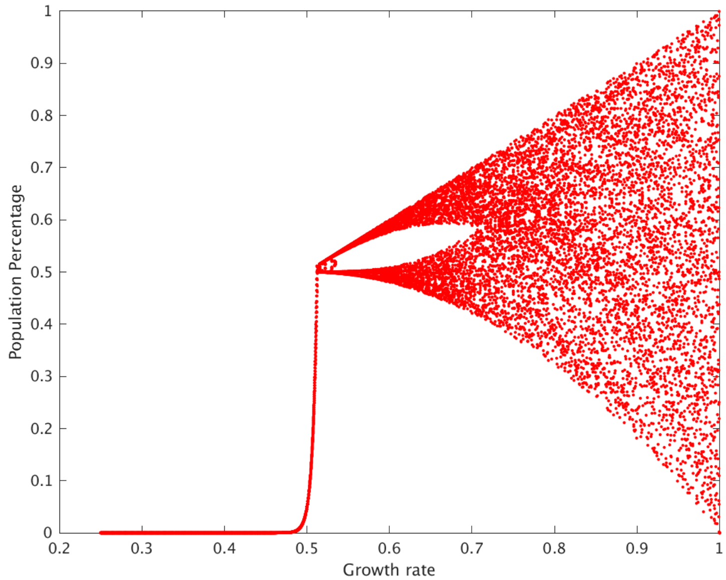
In short: using the given equation $x_{n+1} = r(1-2|x_n - ½|)$, values of r<0.5 will converge to 0. Therefore, the stable point when recycling population values between iterations of the growth rate is 0. This is demonstrated by the below graph:



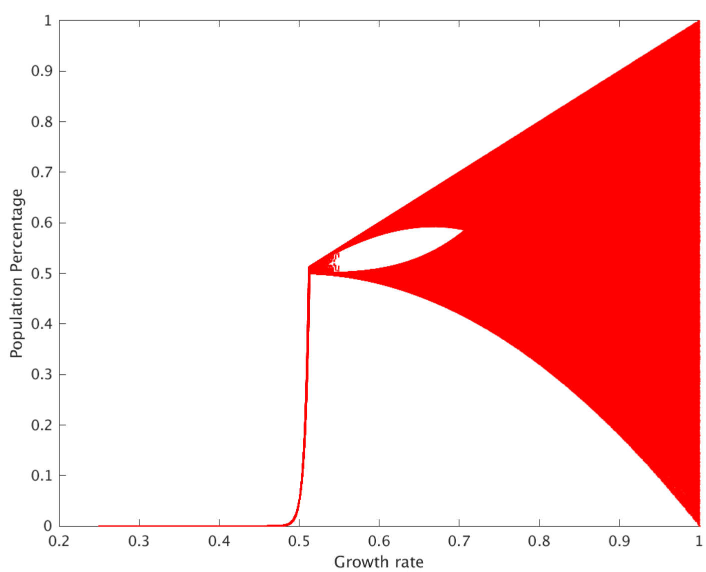
This graph shows all 100 iterations for each growth rate: we can see that approximately the first 10 values at the initial growth rate still have any population. The population rapidly declines into zero, where it will remain. There are several solutions to this problem: a piecewise function for r<0.5, limiting r to r>=0.5, or resetting x for each value of r.

Rac158, R. Aidan Campbell
10-17-2016
EECS131 Quiz 4:

Given that we're attempting to produce a bifurcation diagram of a chaotic function, I opted to reset x for each value of r. In this implementation, instead of recycling x from the previous r value, x is reinitialized to the constant 0.05. From doing this, we produce the following graph:



From this graph we see the single stable point at an r value of 0.5, and a population percentage at 0.5. This is to be expected. The points leftward of 0.5 converge to 0, more rapidly as you move further away. The points rightward of 0.5 follow the expected bifurcation diagram of a tent map, similar to a probability density map. Using finer steps of r, we get the following graph:



Rac158, R. Aidan Campbell
10-17-2016
EECS131 Quiz 4:

On Pascal’s Triangle

Pascal’s triangle is a mathematical phenomenon constructed by summing the element one row above and left, with the element one row above and right. Edges of the triangle are defined to have a value of 1. An example is seen below:

```
>> my_pascal(5)

ans =

     0     0     0     0     1     0     0     0     0
     0     0     0     1     0     1     0     0     0
     0     0     1     0     2     0     1     0     0
     0     1     0     3     0     3     0     1     0
     1     0     4     0     6     0     4     0     1
```

The nature of the problem lends itself to recursion, with edge values of 1 being the base case. With that base case, calculation of the element at location (n, i) is defined to be

`val(n-1, i-1) + val(n-1, i)`

Using this formula, we can produce the values for each element, and interleave zeros as necessary to produce the expected image.