30 de novembro

Motivação

https://www.youtube.com/watch?v=9M7dTmvJxwA

https://www.youtube.com/watch?v=FIKS38DJ444

Começando pelo começo: de onde surge a ideia?

Simular o crescimento de uma população de acordo com uma regra simples.

A regra mais simples seria que a população vai ser dada pela multiplicação do número de indivíduos atual vezes uma taxa de crescimento r

$$x_{n+1} = r \times x_n$$

A principal hipótese aqui seria que:

- Mais animais implica em um mais descendentes

Mas isso faria com que o número de indivíduos aumentasse exponencialmente.

Para adicionar uma restrição a isso podemos adicionar um novo termo:

$$x_{n+1} = rx_n(1 - x_n)$$

A principal hipótese aqui seria que:

- Existe um limite superior para o número de animais que podem habitar o ambiente de forma sustentável

E aqui assumimos que x está normalizado e dentro do intervalo $[0, 1]$.
Conforme x se aproxima do máximo o crescimento da população é limitado.

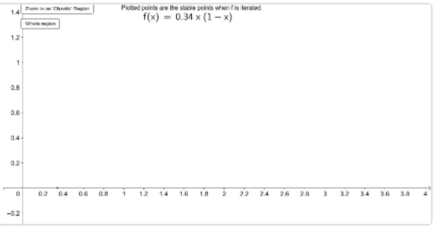
Isso vai definir um mapeamento da população atual para a população do próximo ano, conhecido como **mapa logístico.**

(perdão por usar o geogebra)

Bifurcation Diagram (Logistic Map)

Explore the stable points of the Logistic Map - this is the function (often used as a population model) that first caused the phenomenon of Chaos to ...

https://www.geogebra.org/m/wQbHRgye

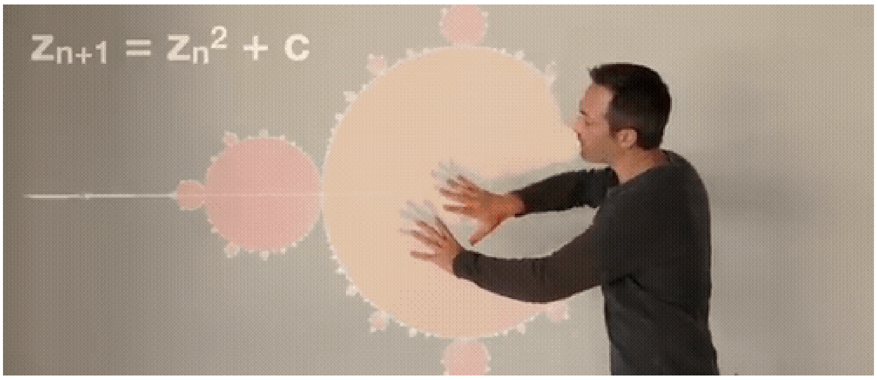


A dobra de período acontece cada vez mais rápido!

Mas em 3.59 algo diferente acontece, o dobramento de período tem que acabar depois desse ponto

▼ Ta mas o que que tem a ver com os fractais lá?

Na verdade o diagrama de bifurcação está contido no plot do conjunto de Mandelbrot:



Eu não vou tentar explicar isso aqui, mas tá tudo nesse vídeo aqui:

https://youtu.be/ovJcsL7vyrk

▼ Beleza mas e o projeto?

De novo recorremos a sabedoria do Heitor, que preparou um ótimo material extra:

fiscomp-2018/projeto5 at master · heitorPB/fiscomp-2018

O código 1.f90 imprime no terminal xi, i, d como pedido nos itens b) e c). O código 1.py é um script Python que compila, executa e faz os gráficos: O código 2.f90 imprime no terminal

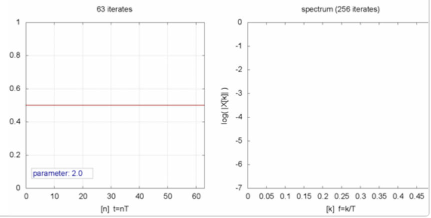
https://github.com/heitorPB/fiscomp-2018/tree/master/projeto5

E como sempre vamos recorrer a famosa wikipedia:

Logistic map - Wikipedia

The logistic map is a polynomial mapping (equivalently, recurrence relation) of degree 2, often reffered as an archetypal example of how complex,

https://en.wikipedia.org/wiki/Logistic_map



De lá a gente tira isso aqui:

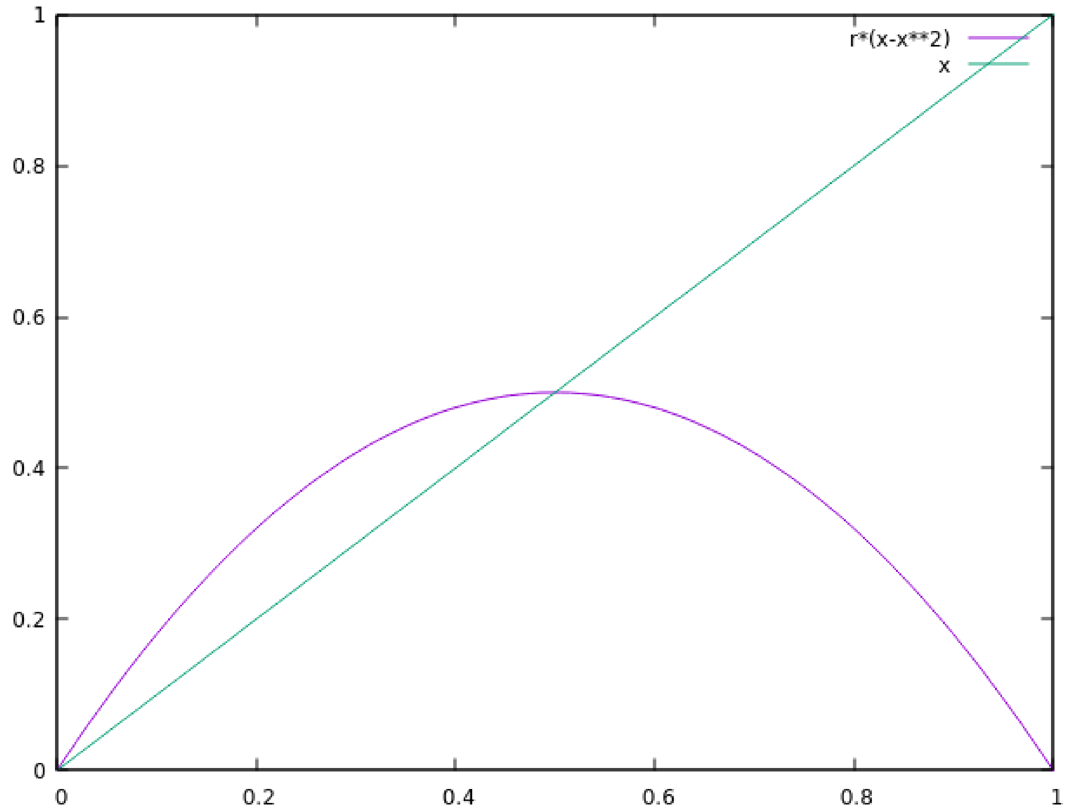
```
import numpy as np
import matplotlib.pyplot as plt

interval = (2.8, 4)  # start, end
accuracy = 0.0001
reps = 600  # number of repetitions
numtoplot = 200
lims = np.zeros(reps)

fig, biax = plt.subplots()
fig.set_size_inches(16, 9)

lims[0] = np.random.rand()
for r in np.arange(interval[0], interval[1], accuracy):
    for i in range(reps-1): lims[i+1] = r*lims[i]*(1-lims[i])
    biax.plot([r]*numtoplot, lims[reps-numtoplot:], 'b.', markersize=.02)

biax.set(xlabel='r', ylabel='x', title='logistic map')
plt.show()
```

Sobre as propriedades de G(x):

Pontos fixos $G(x)$: um mapeamento de um ponto nele mesmo!

$$G(x') = x'$$

▼ Solução gráfica: plotar $x_{n+1} \times x_n$ e a função de identidade

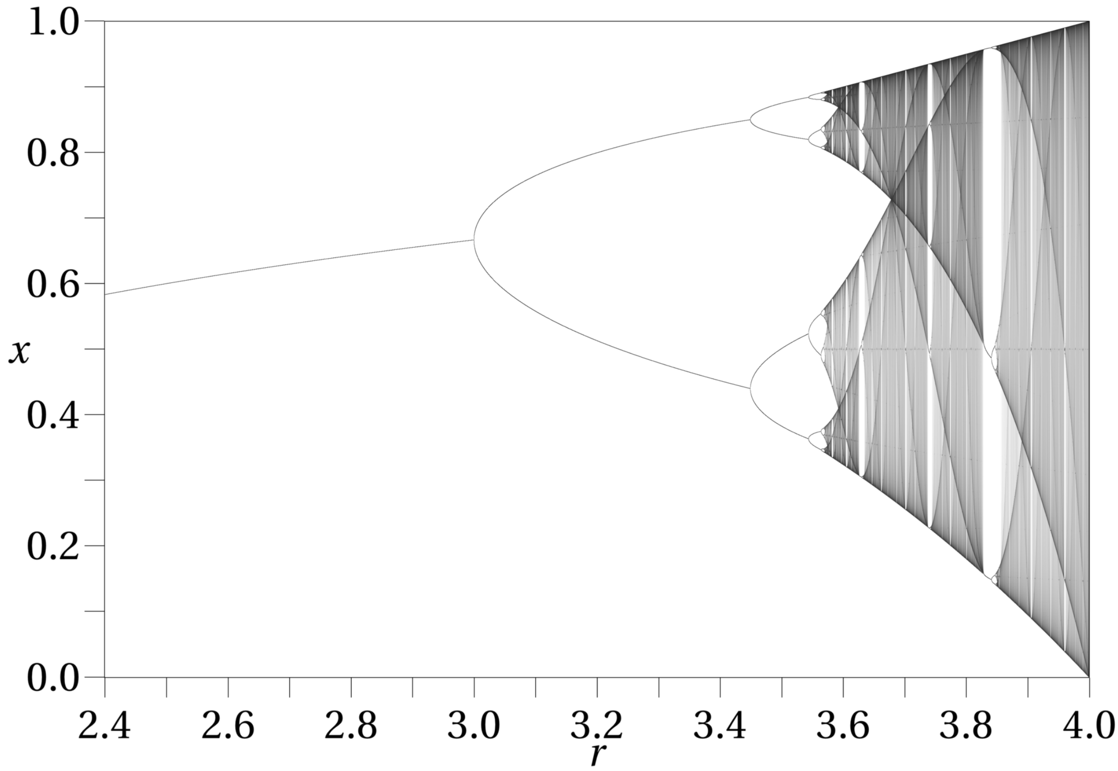


- $x = 0$ é sempre uma solução

$$x_{n+1} = rx_n(1 - x_n)$$
$$x_{n+1} = r * 0(1 - x_n) = 0$$

- r tem que estar entre $[0, 4)$, o que acontece no gráfico abaixo com $r > 4$?

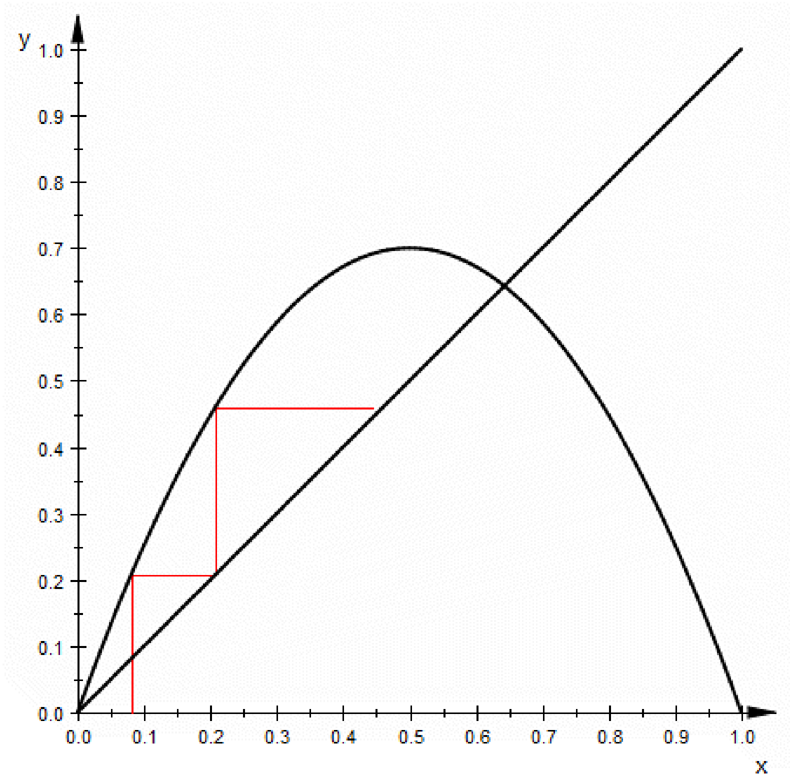
  Fica maior que 1 e menor que zero!



Vamos fazer o impensável então colocar valores maiores que 4 ali e ver o que acontece:

```
import numpy as np
import matplotlib.pyplot as plt

interval = (3.9, 4.1)  # start, end
accuracy = 0.001
reps = 600  # number of repetitions
numtoplot = 200
lims = np.zeros(reps)

fig, biax = plt.subplots()
fig.set_size_inches(16, 9)

lims[0] = np.random.rand()
for r in np.arange(interval[0], interval[1], accuracy):
    for i in range(reps-1): lims[i+1] = r*lims[i]*(1-lims[i]); print(lims[i+1])
    biax.plot([r]*numtoplot, lims[reps-numtoplot:], 'b.', markersize=.02)

biax.set(xlabel='r', ylabel='x', title='logistic map')
plt.show()
```

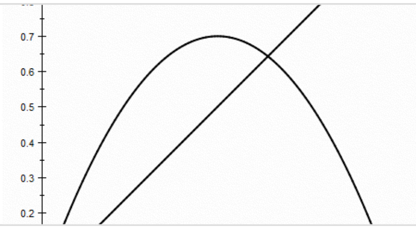
Diagrama de Cobweb



Cobweb plot - Wikipedia

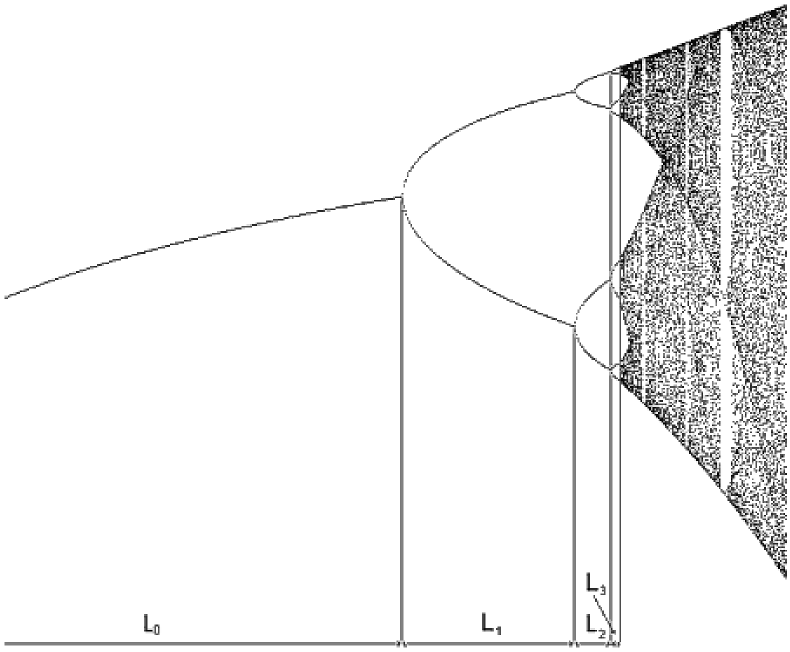
A cobweb plot, or Verhulst diagram is a visual tool used in the dynamical systems field of mathematics to investigate the qualitative behaviour of one-

https://en.wikipedia.org/wiki/Cobweb_plot



Constante de Feigenbaum

Existe alguma relação que podemos extrair nessa dobre de período?



$$\delta = \frac{r_2 - r_1}{r_3 - r_2} = 4.669201609102990671853203820466$$

▼ Conteúdo Extra e Dicas

O wolfram é seu amigo!!!

logistic map - Wolfram|Alpha

Uh oh! Wolfram|Alpha doesn't run without JavaScript. Please enable JavaScript. If you don't know how, you can find instructions . Once you've done that, refresh

https://www.wolframalpha.com/input?i=logistic+map&assumption=%7B%22F%22%2C+%22LogisticMap%22%2C+%22x0%22%7D+-%3E%220.1%22&assumption=%7B%22C%22%2C+%22logistic+map%22%7D+-%3E+%7B%22Formula%22%7D&assumption=%7B%22F%22%2C+%22LogisticMap%22%2C+%22r%22%7D+-%3E%224%22

Cobweb plot - Wolfram|Alpha

Uh oh! Wolfram|Alpha doesn't run without JavaScript. Please enable JavaScript. If you don't know how, you can find instructions . Once you've done that, refresh

https://www.wolframalpha.com/input?i=Cobweb+plot

The Feigenbaum Constant:

The Feigenbaum Constant (4.669) - Numberphile

Binge on learning at The Great Courses Plus: http://ow.ly/Z5yR307LfxY The Feigenbaum Constant and Logistic Map - featuring Ben Sparks.

https://www.youtube.com/watch?v=ETrYE4MdoLQ