

Ordem e Caos

Ana Alexandra Dias, Gonçalo
Torres, Issufo Carlitos
Antônio, Machado António

Indíce

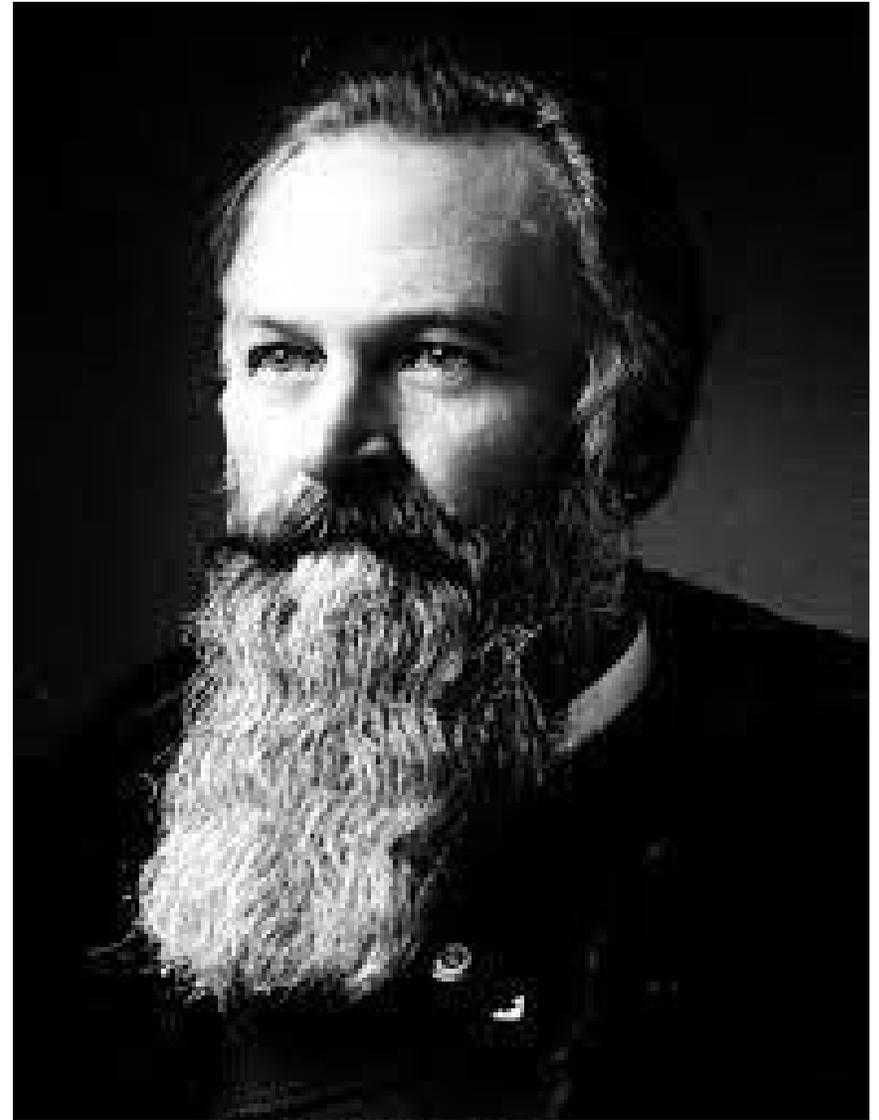
- Objetivos do projeto
- Introdução
- Diferença entre sistemas caóticos e não caóticos
- Simulações de sistemas caóticos e não caóticos
- Conclusão

Objetivos do projeto

- Mostrar a diferença entre sistemas caóticos e sistemas não caóticos
- Utilizar programação para realizar simulações em 3D de sistemas caóticos e sistemas não caóticos

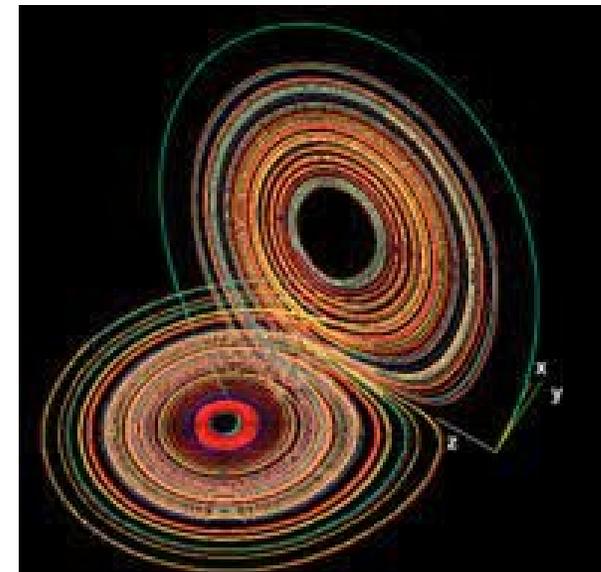
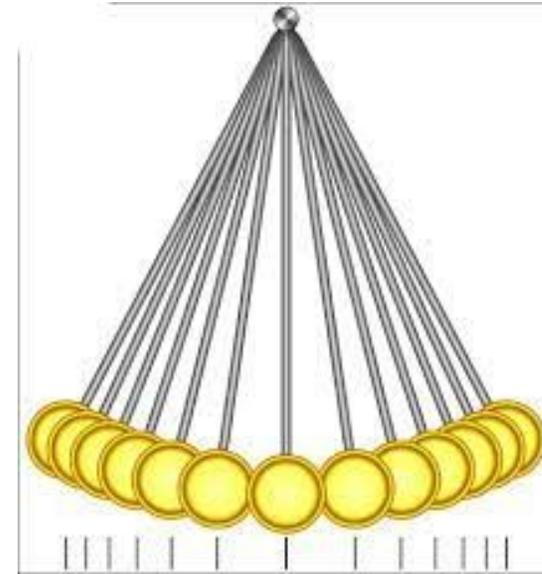
Introdução

- Em 1955, ao testar um programa de computador que simulava massas de ar, Edward Lorenz verificou que uma alteração insignificante nos valores iniciais de temperatura, pressão e humidade conduziram a resultados muito diferentes dos iniciais.
- Para Lorenz era como se “o bater das asas de uma borboleta no Brasil causasse, tempos depois, um tornado no Texas”. Esta instabilidade nos sistemas complexos ficou conhecida como “efeito borboleta”.



Qual a diferença entre sistemas caóticos e não caóticos?

- Sistema não caótico
 - Exemplos: pêndulo, queda de um grave, Lua
- Sistema caótico
 - Exemplos: clima, Atrator de Lorenz



Sistema não caótico

Bolas saltitantes

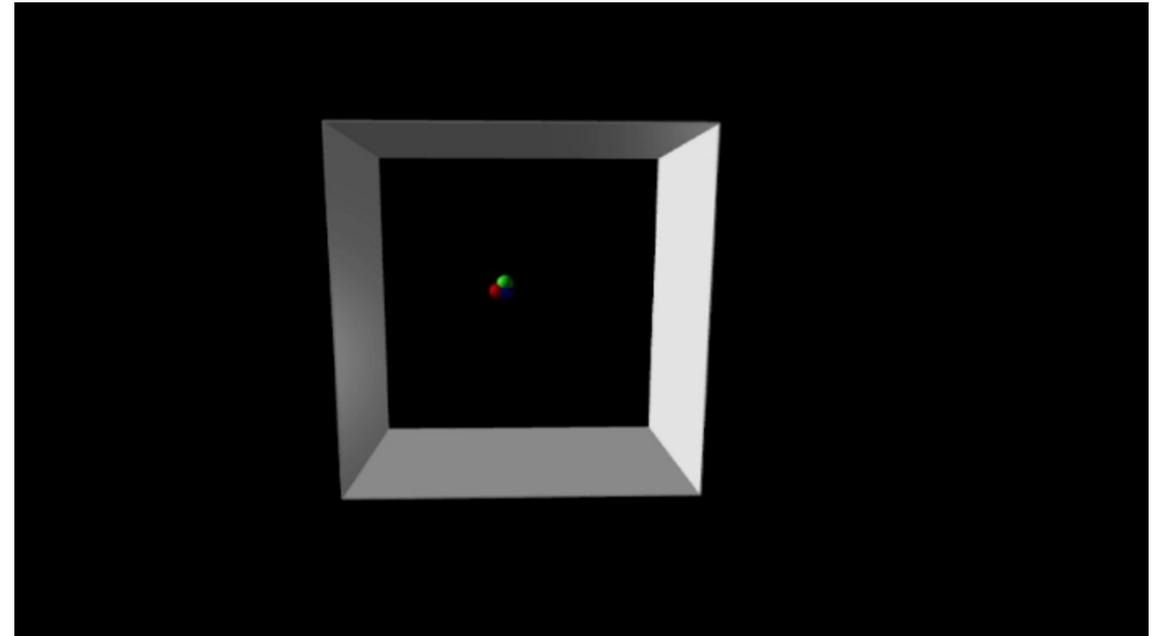
```
1 GlowScript 3.1 VPython
2 """
3 arrow(axis=vector(1,0, 0), color=color.red)
4 arrow(axis=vector(0,1, 0), color=color.green)
5 arrow(axis=vector(0,0, 1), color=color.blue)
6 """
7
8
9 r=vector(18,0,0)
10 rec=box(pos=r, length=48.0, height=0.6, width=2)
11
12
13 y0=3
14 v0y=6.0
15 v10y=6.1
16 a=-9.8
17 v0x=2.0
18
19
20 li=[]
21 bola1=sphere(color=color.blue, pos=vector(1,y0,1), radius=0.3, velocity=vector(2, v0y, 0), make_trail=True)
22 bola2=sphere(color=color.red, pos=vector(1,y0,1), radius=0.3, velocity=vector(2, v10y, 0), make_trail=True)
23 li.append(bola1)
24 li.append(bola2)
25
26 dt=0.01
27 t=0
28 while t<20:
29     t=t+dt
30     rate(200)
31
32     for i in range(2):
33         li[i].pos.y=li[i].pos.y+li[i].velocity.y*dt+0.5*a*dt**2
34         li[i].velocity.y=li[i].velocity.y+a*dt
35         li[i].pos.x=li[i].pos.x+li[i].velocity.x*dt
36
37
38         if li[i].pos.y<0:
39             li[i].velocity.y=-li[i].velocity.y
```



Sistema não caótico

Bolas na caixa

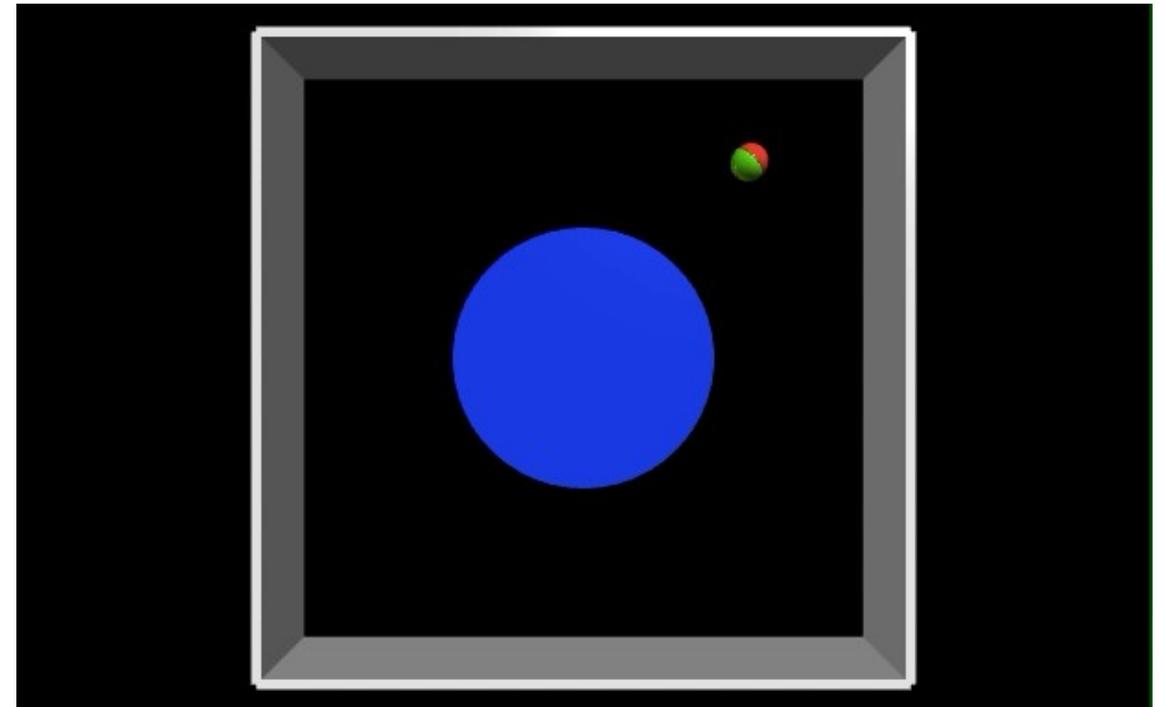
```
1 GlowScript 3.1 VPython
2 #CAIXA COM 3 bolas
3 #arrow(axis=vector(0,0,1), color=color.blue)
4 #arrow(axis=vector(0,1,0), color=color.green)
5 #arrow(axis=vector(1,0,0), color=color.red)
6 rec1=box(pos=vector(0,1,0), length=1, height=0.01, width=2.0)
7 rec2=box(pos=vector(0,0,1), length=1, height=2, width=0.01)
8 rec3=box(pos=vector(0,0,-1), length=1, height=2, width=0.01)
9 rec4=box(pos=vector(0,-1,0), length=1, height=0.01, width=2.0)
10 x0=0
11 y0=0.1
12 z0=0.1
13 v0y=1.5
14 v0z=0.5
15 a=0
16 bola1=sphere(color=color.blue, pos=vector(x0,y0,z0), radius=0.05, velocity=vector(0,v0y,v0z), make_trail=True)
17 bola2=sphere(color=color.green, pos=vector(x0,1.5*y0,z0), radius=0.05, velocity=vector(0,v0y,v0z), make_trail=True)
18 bola3=sphere(color=color.red, pos=vector(x0,y0,1.5*z0), radius=0.05, velocity=vector(0,v0y,v0z), make_trail=True)
19 dt=0.01
20 t=0
21 li=[]
22 li.append(bola1)
23 li.append(bola2)
24 li.append(bola3)
25 r=[0.05,0.05,0.05]
26 while t<5:
27     t=t+dt
28     rate(100)
29 while t<20:
30     for i in range(3):
31         li[i].pos.y=li[i].pos.y+li[i].velocity.y*dt
32         li[i].pos.z=li[i].pos.z+li[i].velocity.z*dt
33         if li[i].pos.y<-1+r[i] or li[i].pos.y>1-r[i]:
34             li[i].velocity.y=-li[i].velocity.y
35         if li[i].pos.z<-1+r[i] or li[i].pos.z>1-r[i]:
36             li[i].velocity.z=-li[i].velocity.z
37     t=t+dt
38     rate(100)
```



Sistema caótico

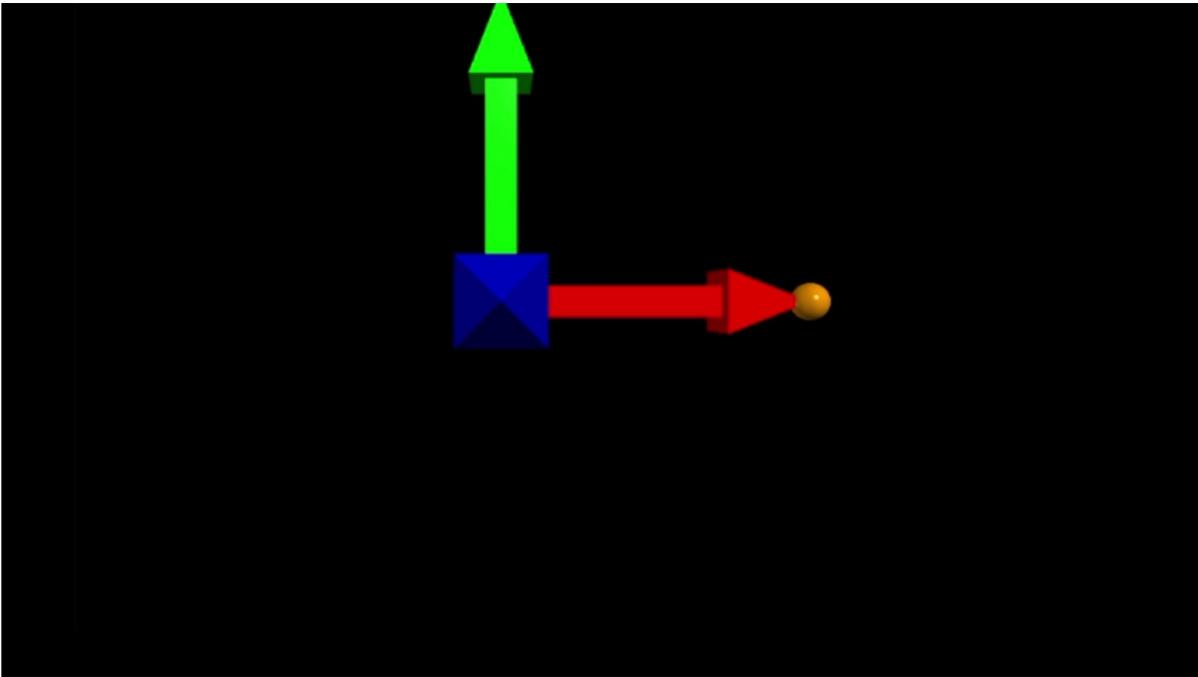
Bilhar com círculo no meio

```
7 b1=box(pos=vector(1,0,0), length=0.03, height=2, width=0.3)
8 b2=box(pos=vector(-1,0,0), length=0.03, height=2, width=0.3)
9 b3=box(pos=vector(0,1,0), length=2, height=0.03, width=0.3)
10 b4=box(pos=vector(0,-1,0), length=2, height=0.03, width=0.3)
11
12 x0=0
13 y0=0
14 z0=-0.1
15 w=0.5
16 r=0.4
17
18 c=cylinder(pos=vector(x0,y0,z0), axis=vector(0,0,1), size=vector(w/2,2*r,2*r), color=color.blue)
19
20 y10=0.6000006
21 x10=0.50000051
22 v10y=1.51
23 v10x=1.51
24 y20=0.60000061
25 x20=0.5000005
26 v20y=1.5
27 v20x=1.5
28 x30=0.50000052
29 y30=0.60000062
30 v30y=1.52
31 v30x=1.52
32 y40=0.60000063
33 x40=0.50000053
34 v40y=1.53
35 v40x=1.53
36 y50=0.60000064
37 x50=0.50000054
38 v50y=1.54
39 v50x=1.54
40 y60=0.60000065
41 x60=0.50000055
42 v60y=1.55
43 v60x=1.55
44
45 li=[]
46 bola1=sphere(color=color.blue, pos=vector(x10,y10,0), radius=0.06, velocity=vector(v10x, v10y, 0), make_trail=True)
47 bola2=sphere(color=color.green, pos=vector(x20,y20,0), radius=0.06, velocity=vector(v20x, v20y, 0), make_trail=True)
48 bola3=sphere(color=color.orange, pos=vector(x30,y30,0), radius=0.06, velocity=vector(v30x, v30y, 0), make_trail=True)
49 bola4=sphere(color=color.yellow, pos=vector(x40,y40,0), radius=0.06, velocity=vector(v40x, v40y, 0), make_trail=True)
50 bola5=sphere(color=color.purple, pos=vector(x50,y50,0), radius=0.06, velocity=vector(v50x, v50y, 0), make_trail=True)
51 bola6=sphere(color=color.red, pos=vector(x60,y60,0), radius=0.06, velocity=vector(v60x, v60y, 0), make_trail=True)
52 li.append(bola1)
53 li.append(bola2)
54 li.append(bola3)
```



Sistema caótico

Atrator de Lorenz



bola1=(1, 0, 0)

bola2=(1.00000001, 0, 0)

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x), & \frac{dz}{dt} &= xy - \beta z. \\ \frac{dy}{dt} &= x(\rho - z) - y,\end{aligned}$$

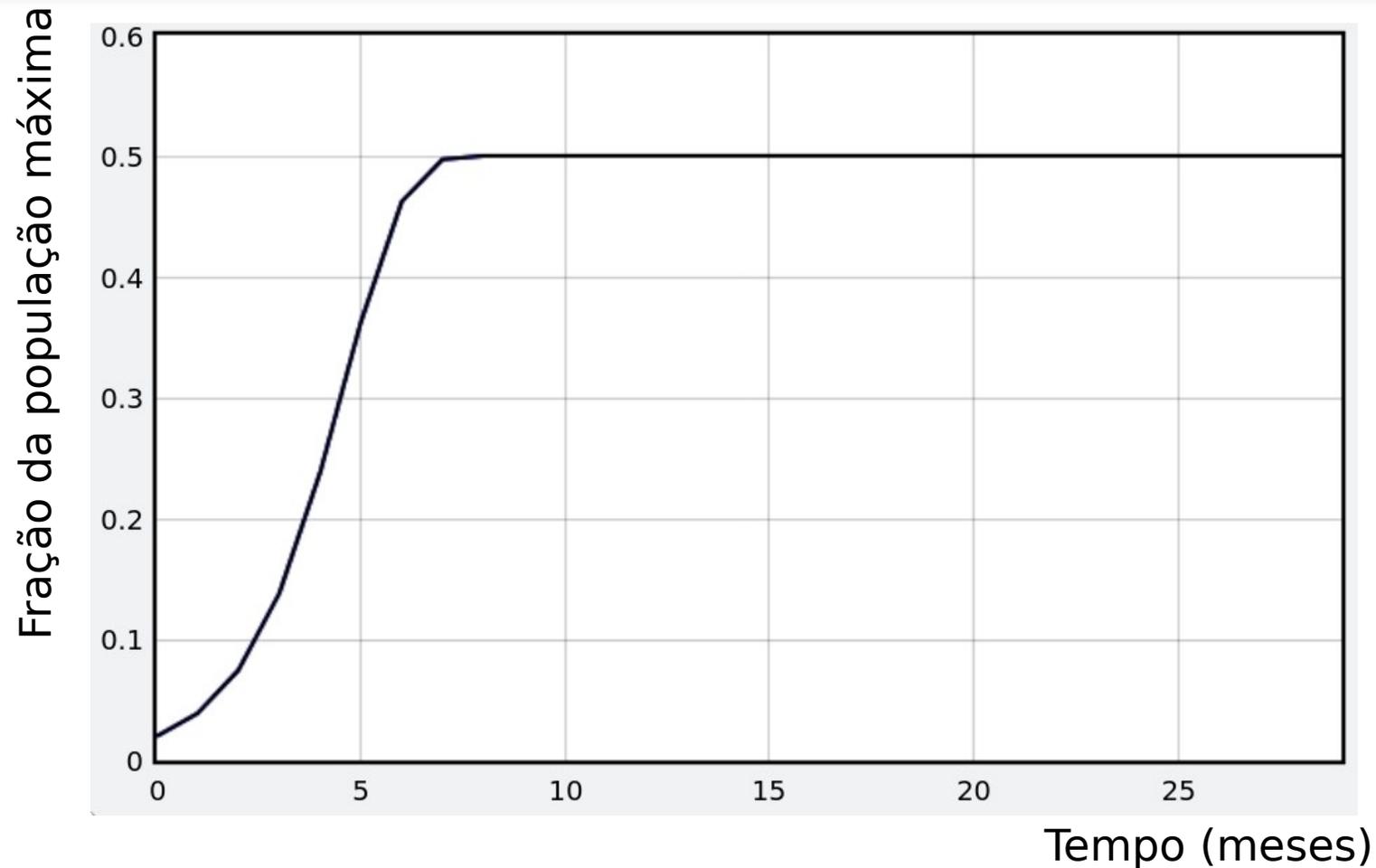
População de coelhos

$r=2$

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

$x=0.01$

$y=0.0100001$



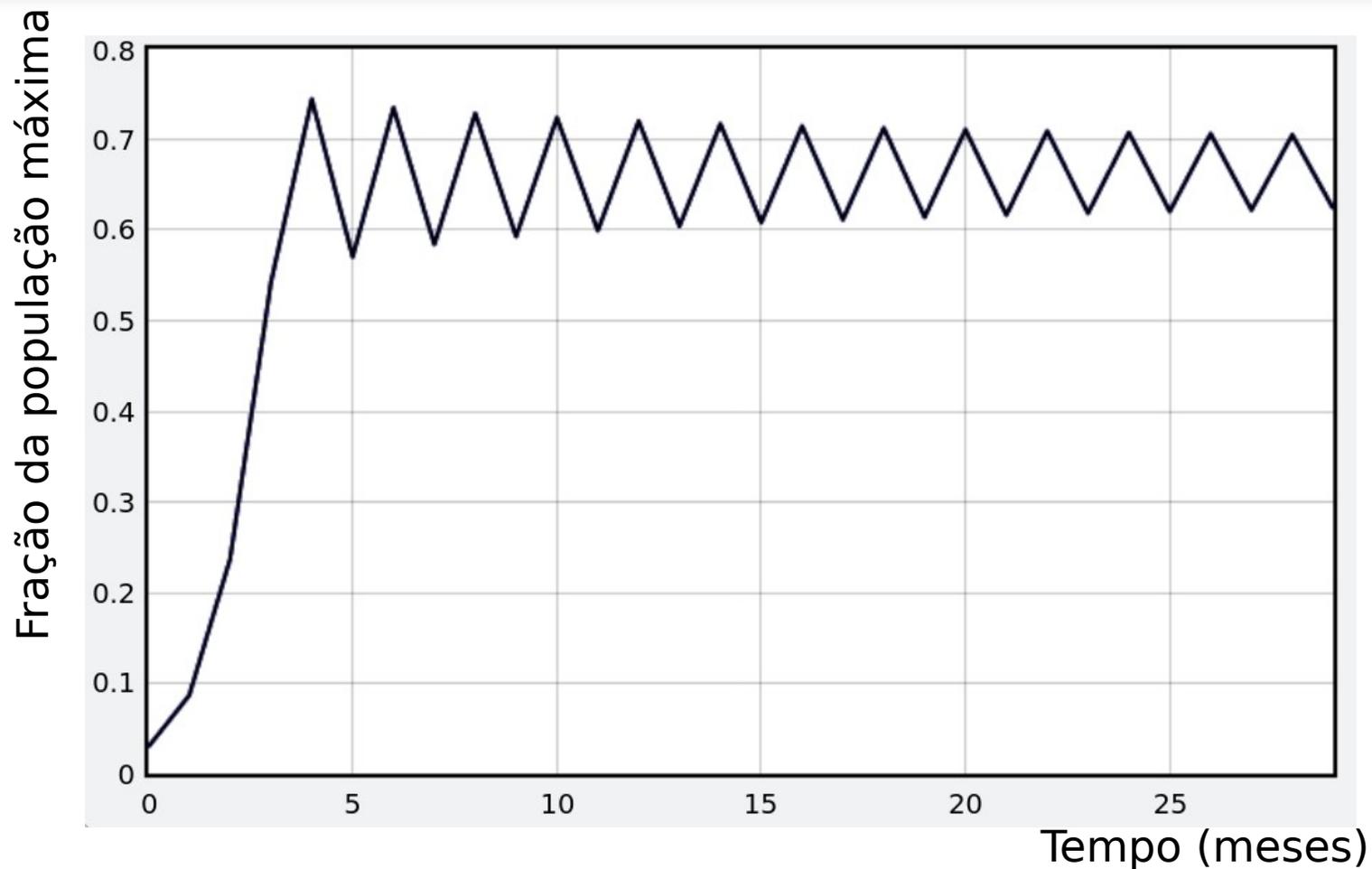
População de coelhos

$r=3$

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

$x=0.01$

$y=0.0100001$



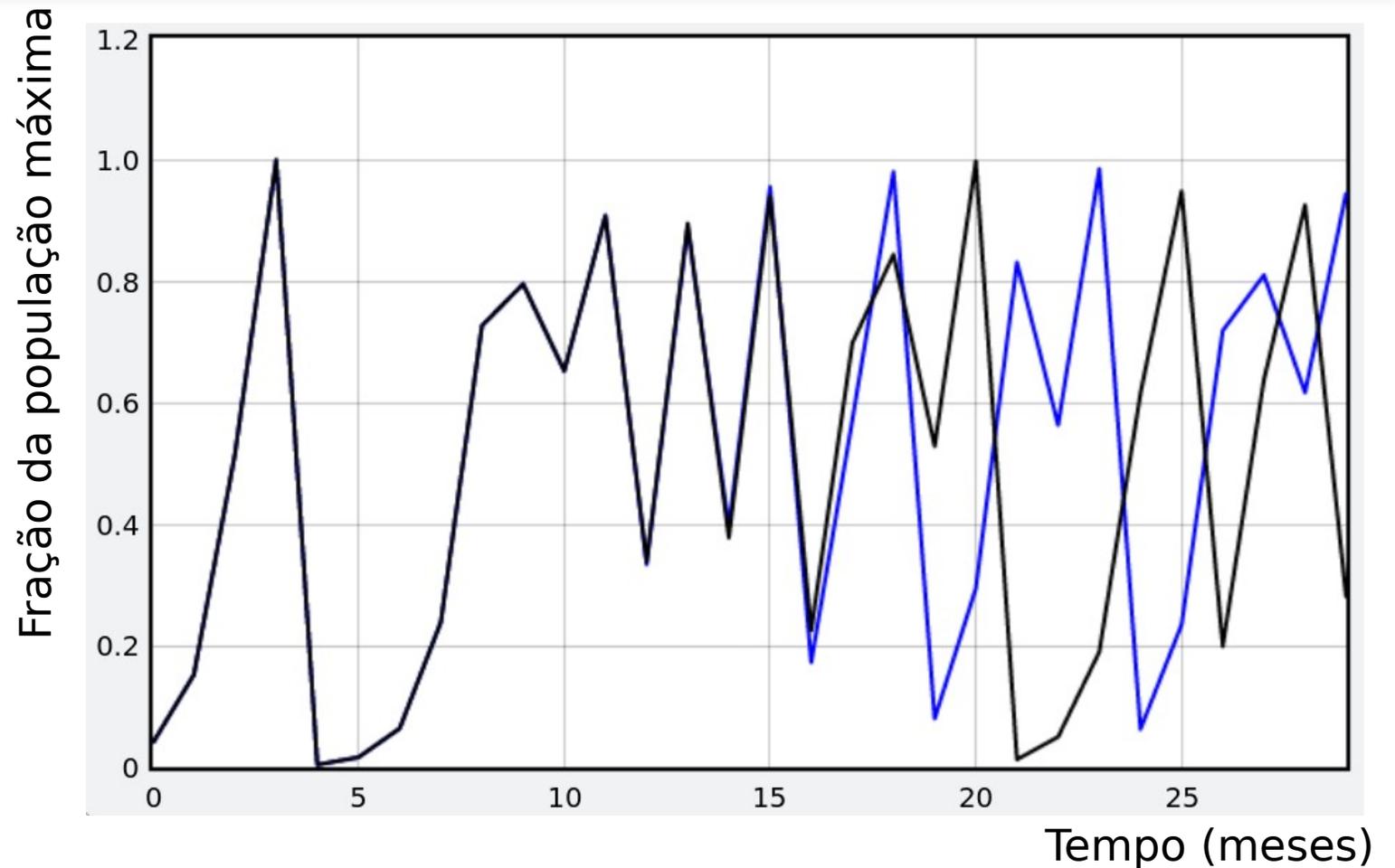
População de coelhos

$r=4$

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

$x=0.01$

$y=0.0100001$



Em síntese:

- Um **sistema não caótico** e um **sistema caótico** são sistemas deterministas. Contudo, enquanto num sistema não caótico uma pequena alteração das condições iniciais não provoca grandes efeitos nas condições finais, num sistema caótico poderá causar uma grande disparidade.