Partial Differential Equations

COS 323

Last time

- More methods for initial value problems
  - Stiff ODEs
  - Backward Euler
  - Multi-step methods
    - Adams methods
- Boundary value problems
  - Definition
  - Shooting method
  - Finite difference method
  - Collocation method

Today

- Finite difference approximations
- Review of finite differences for ODE BVPs
- PDEs
- Phase diagrams
- Chaos

Finite difference approximations

- Given smooth function $f: \mathbb{R} \to \mathbb{R}$, we wish to approximate its first and second derivatives at point x
- Consider Taylor series expansions

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \cdots \\ f(x-h) &= f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(x)}{6}h^3 + \cdots \end{aligned}$$

- Solving for $f'(x)$ in first series, obtain *forward difference approximation*

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(x)}{2}h + \cdots \approx \frac{f(x+h) - f(x)}{h}$$

which is first-order accurate since dominant term in remainder of series is $\mathcal{O}(h)$

- Similarly, from second series derive *backward difference approximation*

$$
\begin{aligned}
f'(x) &= \frac{f(x) - f(x-h)}{h} + \frac{f''(x)}{2}h + \cdots \\
&\approx \frac{f(x) - f(x-h)}{h}
\end{aligned}
$$

which is also first-order accurate

- Subtracting second series from first series gives *centered difference approximation*

$$
\begin{aligned}
f'(x) &= \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(x)}{6}h^2 + \cdots \\
&\approx \frac{f(x+h) - f(x-h)}{2h}
\end{aligned}
$$

which is second-order accurate

- Adding both series together gives *centered difference approximation* for second derivative

$$
\begin{aligned}
f''(x) &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{f^{(4)}(x)}{12}h^2 + \cdots \\
&\approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}
\end{aligned}
$$

  which is also second-order accurate

Finite Difference Method for ODE BVPs

Finite Difference Method

- Introduce mesh points along independent variable
- Replace all derivatives in ODE with finite difference approximations

For example, to solve two-point BVP

$$u'' = f(t, u, u'), \qquad a < t < b$$

with BC

$$u(a) = \alpha, \qquad u(b) = \beta$$

we introduce mesh points $t_i = a + ih$, $i = 0, 1, \ldots, n+1$, where $h = (b-a)/(n+1)$

For example, to solve two-point BVP

$$u'' = f(t, u, u'), \qquad a < t < b$$

with BC

$$u(a) = \alpha, \qquad u(b) = \beta$$

we introduce mesh points $t_i = a + ih$, $i = 0, 1, \ldots, n+1$, where $h = (b-a)/(n+1)$

We replace derivatives by finite difference approximations such as

$$u'(t_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}$$

$$u''(t_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

This yields system of equations

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f\left(t_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\right)$$

to be solved for unknowns y_i, $i = 1, \ldots, n$

Another example: Dissipation of heat from long, thin bar

$$\frac{d^2T}{dx^2} = c(T_a - T) = 0$$

$$T(0) = T_1, \quad T(L) = T_2 \text{ (ends of bar held at fixed T)}$$

$$c = 0.01,\ T_a = 20, \quad T(0) = 40, \quad T(10) = 200$$

Divided differences:

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} - c(T_i - T_a) = 0$$

$$-T_{i-1} + (2 + c\Delta x^2)T_i - T_{i+1} = c\Delta x^2 T_a$$

System of equations

$$-T_{i-1} + (2 + c\Delta x^2)T_i - T_{i+1} = c\Delta x^2 T_a$$

Using 4 interior nodes with $\Delta x = 2$:

$$\begin{bmatrix} -2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{bmatrix}$$

$$T^T = \lfloor 65.9698 \quad 93.7785 \quad 124.5382 \quad 159.4795 \rfloor$$

PDEs

Review: Collocation for ODEs

- *Collocation method* approximates solution to BVP by finite linear combination of basis functions
- For two-point BVP

$$u'' = f(t, u, u'), \qquad a < t < b$$

with BC

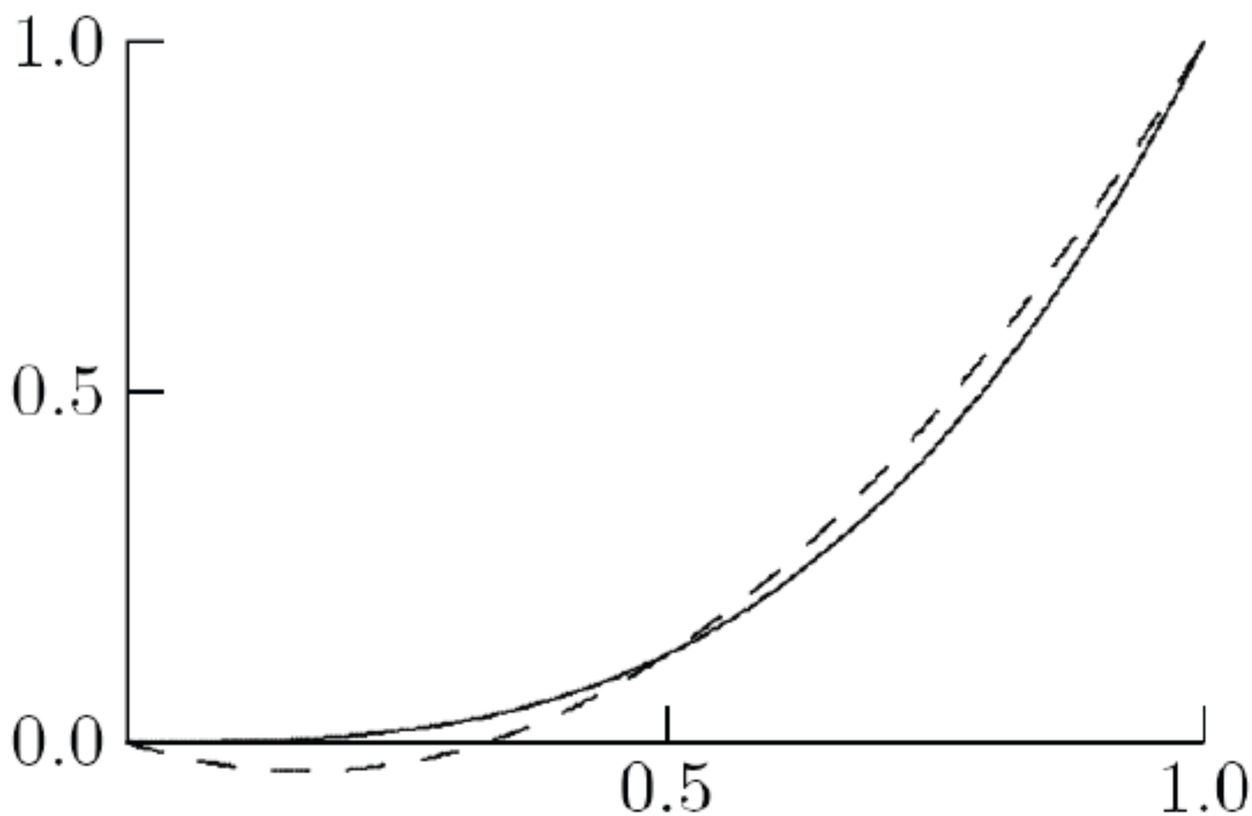
$$u(a) = \alpha, \qquad u(b) = \beta$$

we seek approximate solution of form

$$u(t) \approx v(t, \boldsymbol{x}) = \sum_{i=1}^{n} x_i \phi_i(t)$$

where ϕ_i are basis functions defined on $[a, b]$ and $\boldsymbol{x}$ is n-vector of parameters to be determined

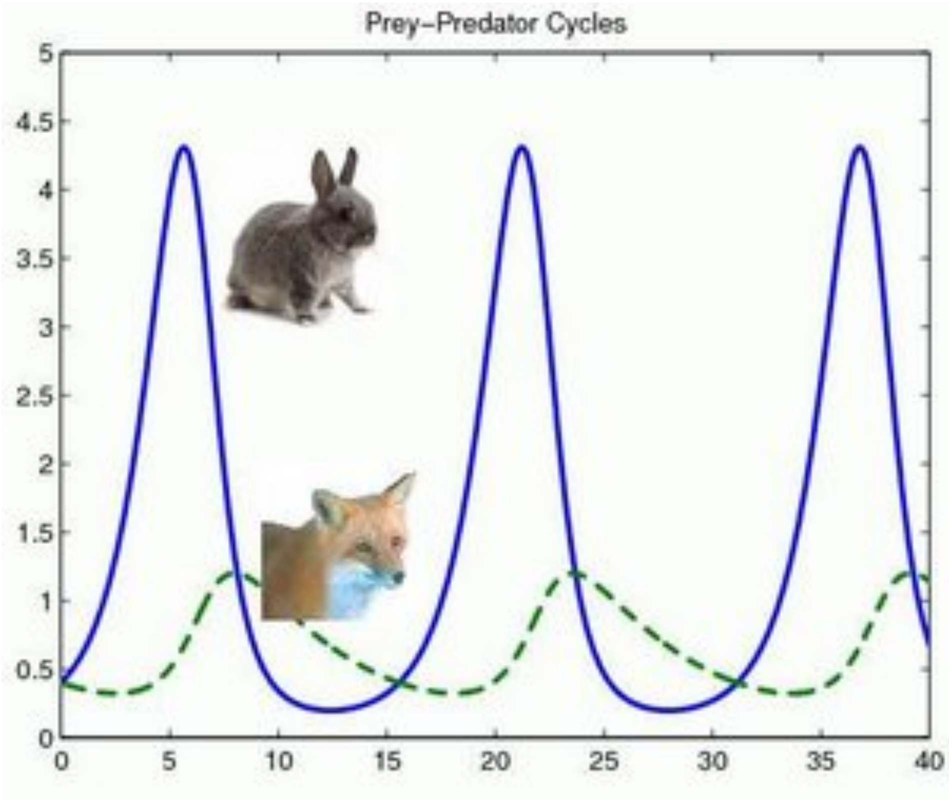
- To determine vector of parameters $\boldsymbol{x}$, define set of n *collocation points*, $a = t_1 < \cdots < t_n = b$, at which approximate solution $v(t, \boldsymbol{x})$ is forced to satisfy ODE and boundary conditions

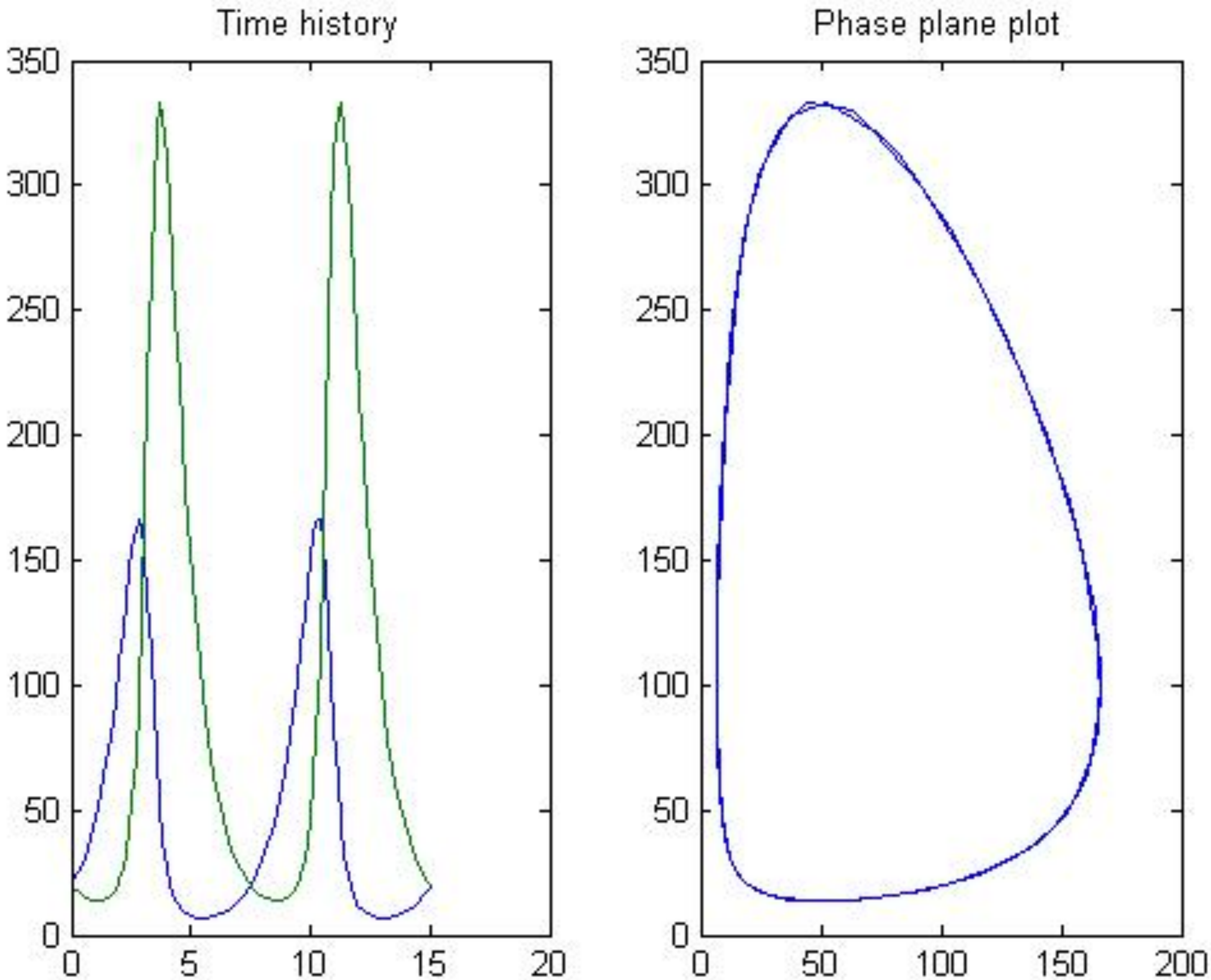


Phase plane diagrams and chaos

Predator-Prey model

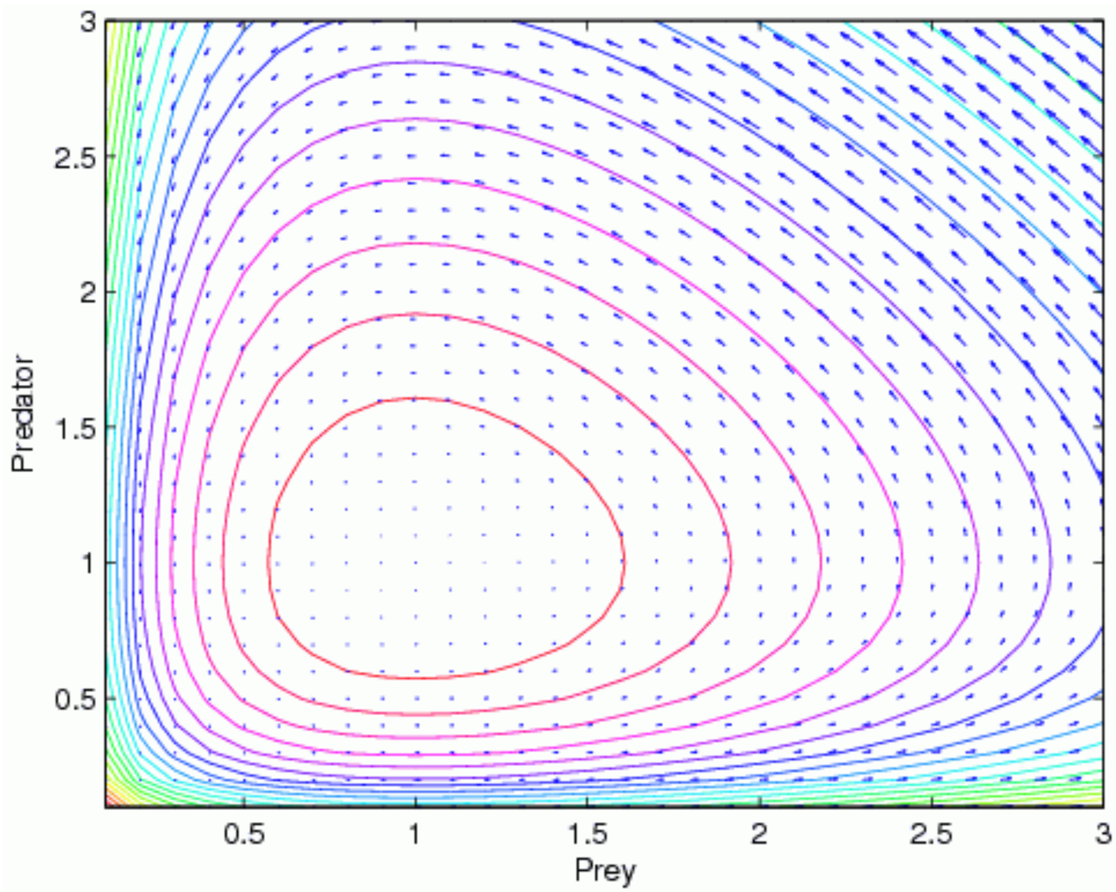
- Lotka-Volterra: $\dfrac{dx}{dt} = ax - bxy, \quad \dfrac{dy}{dt} = -cy + dxy$



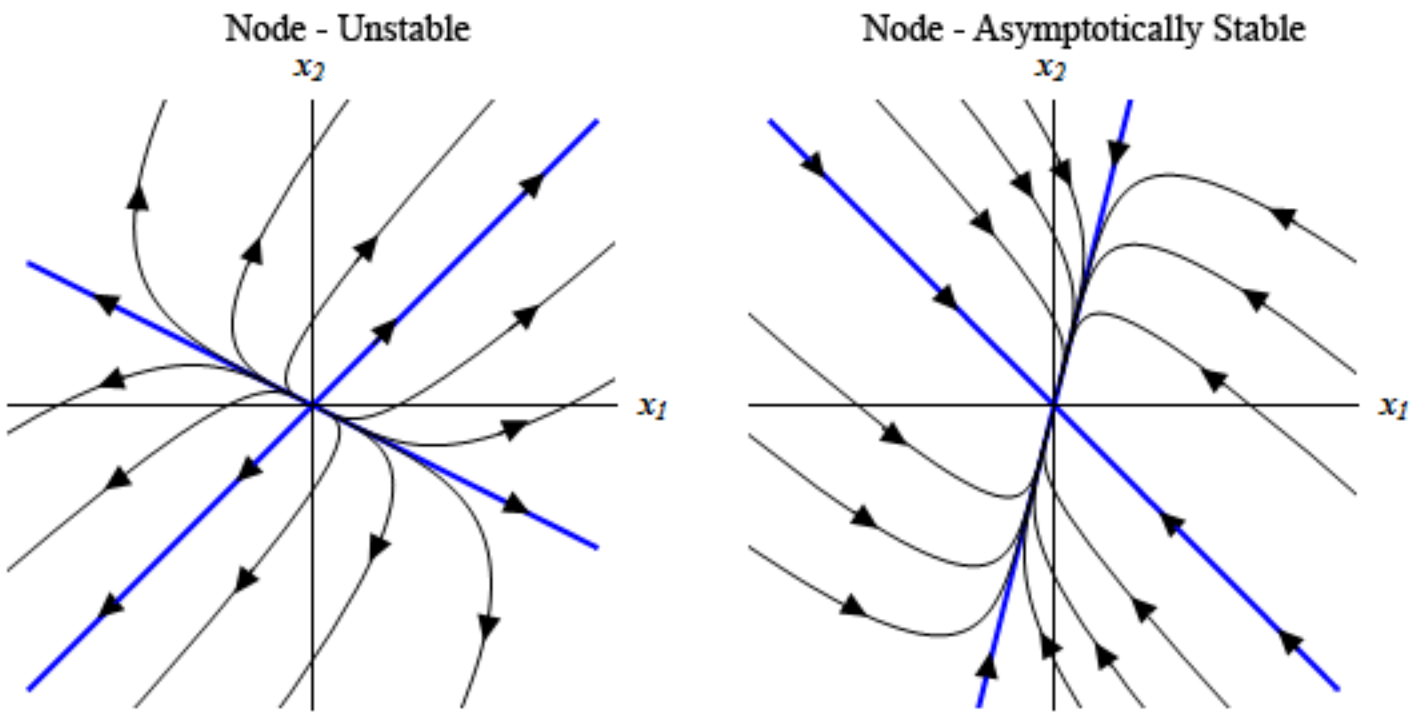


Time history
Phase plane plot
350
300
250
200
150
100
50
0
0
5
10
15
20
0
50
100
150
200

Phase Plane Diagram

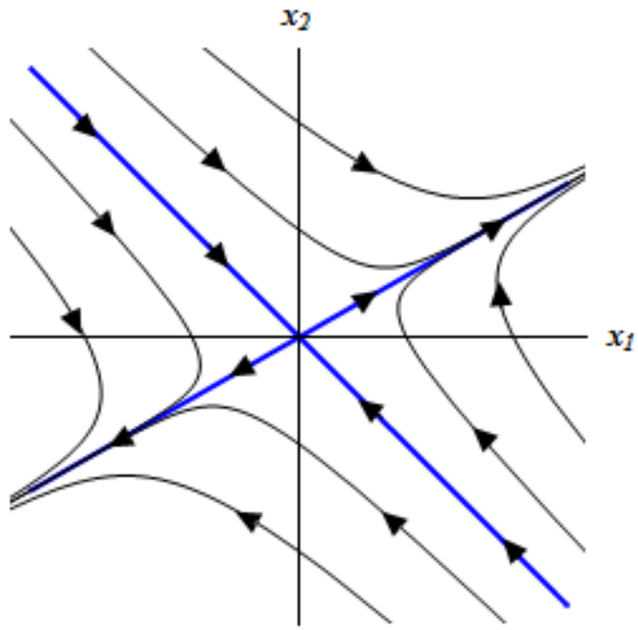


Other possible behaviors

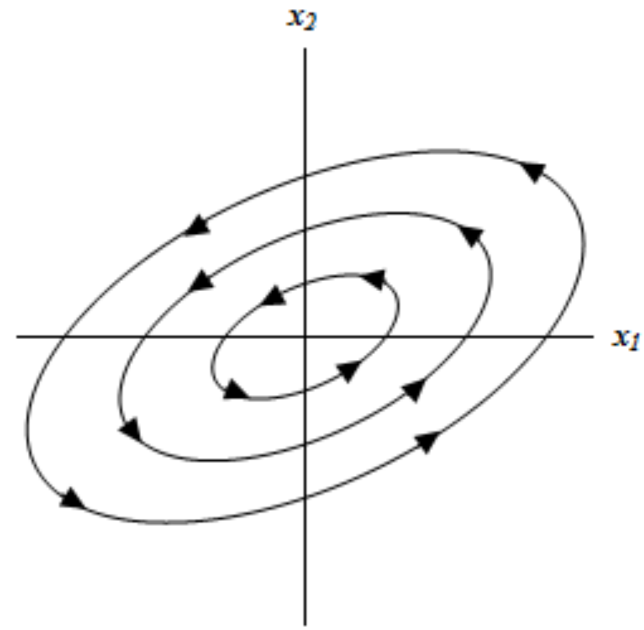


From http://tutorial.math.lamar.edu/classes/de/phaseplane.aspx

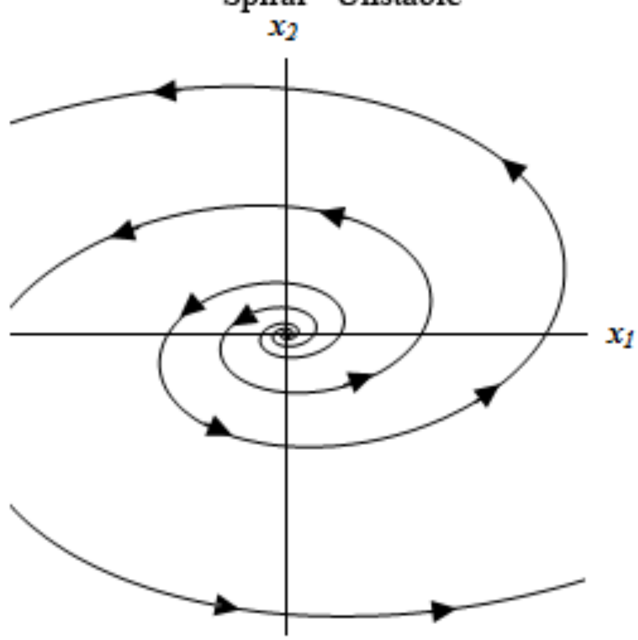
Saddle Point - Unstable



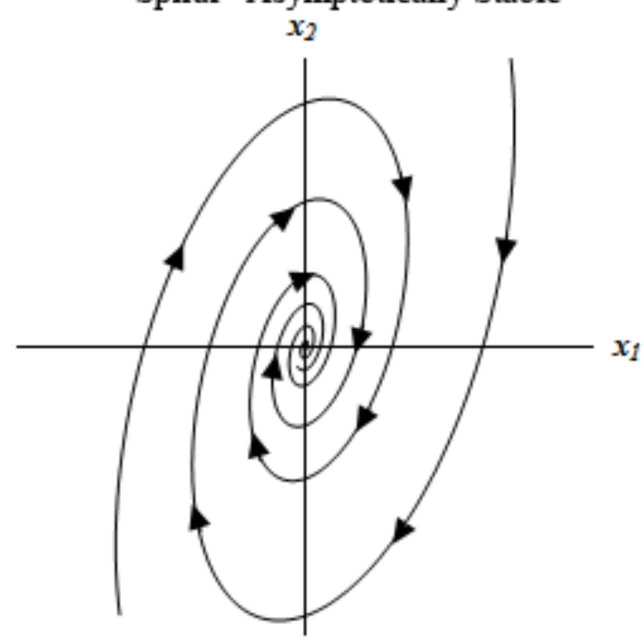
Center - Stable



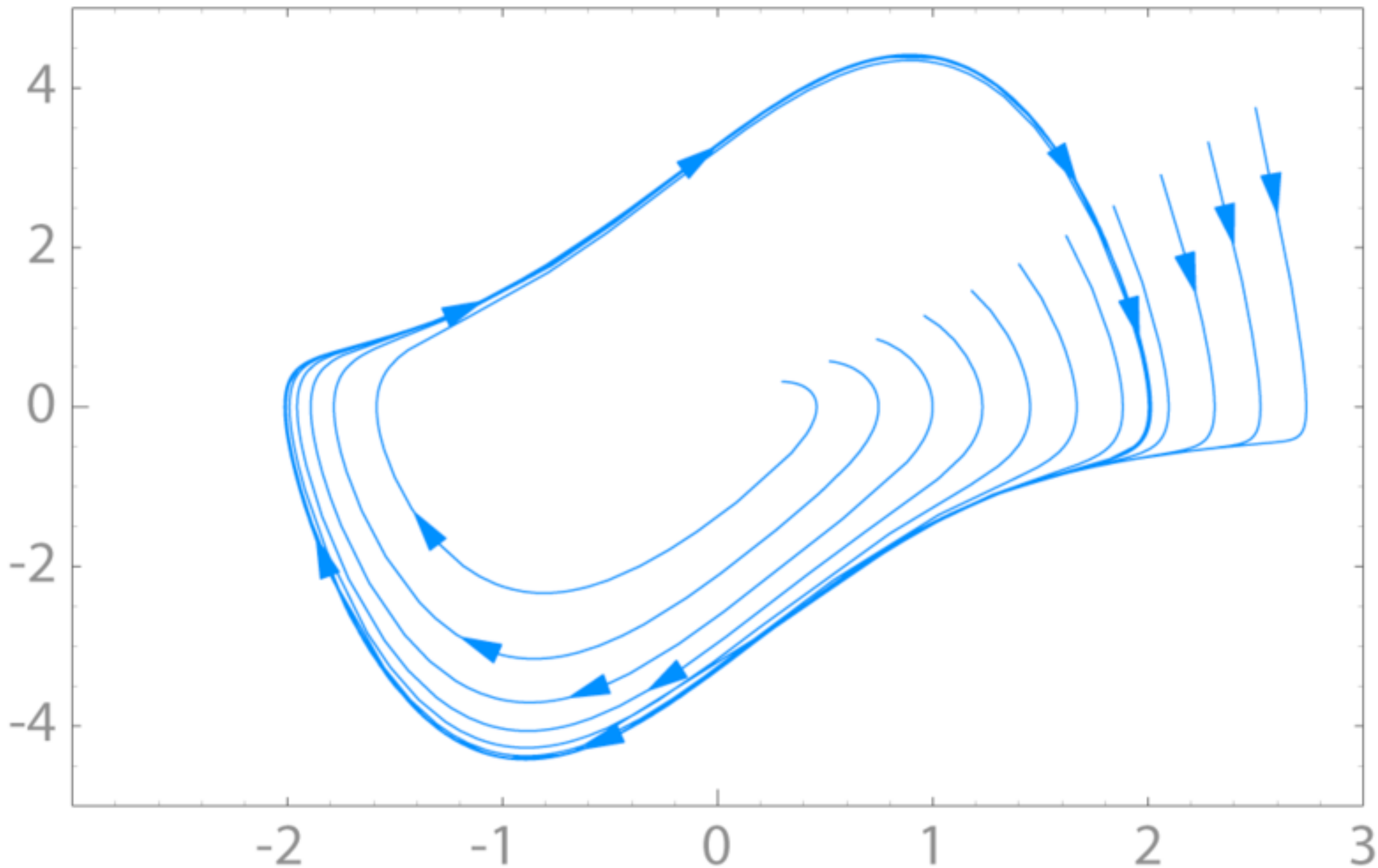
Spiral - Unstable



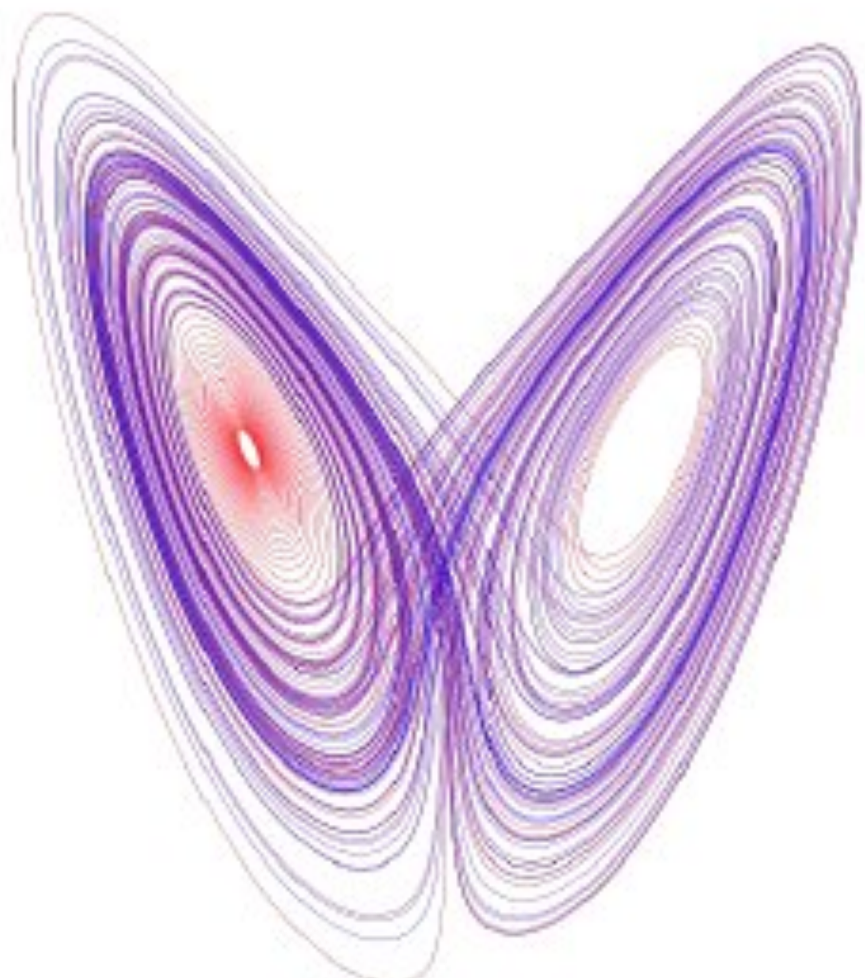
Spiral - Asymptotically Stable



Limit Cycle



Chaos



For more information

- http://cazelais.disted.camosun.bc.ca/262/phaseplane.pdf

Chaos

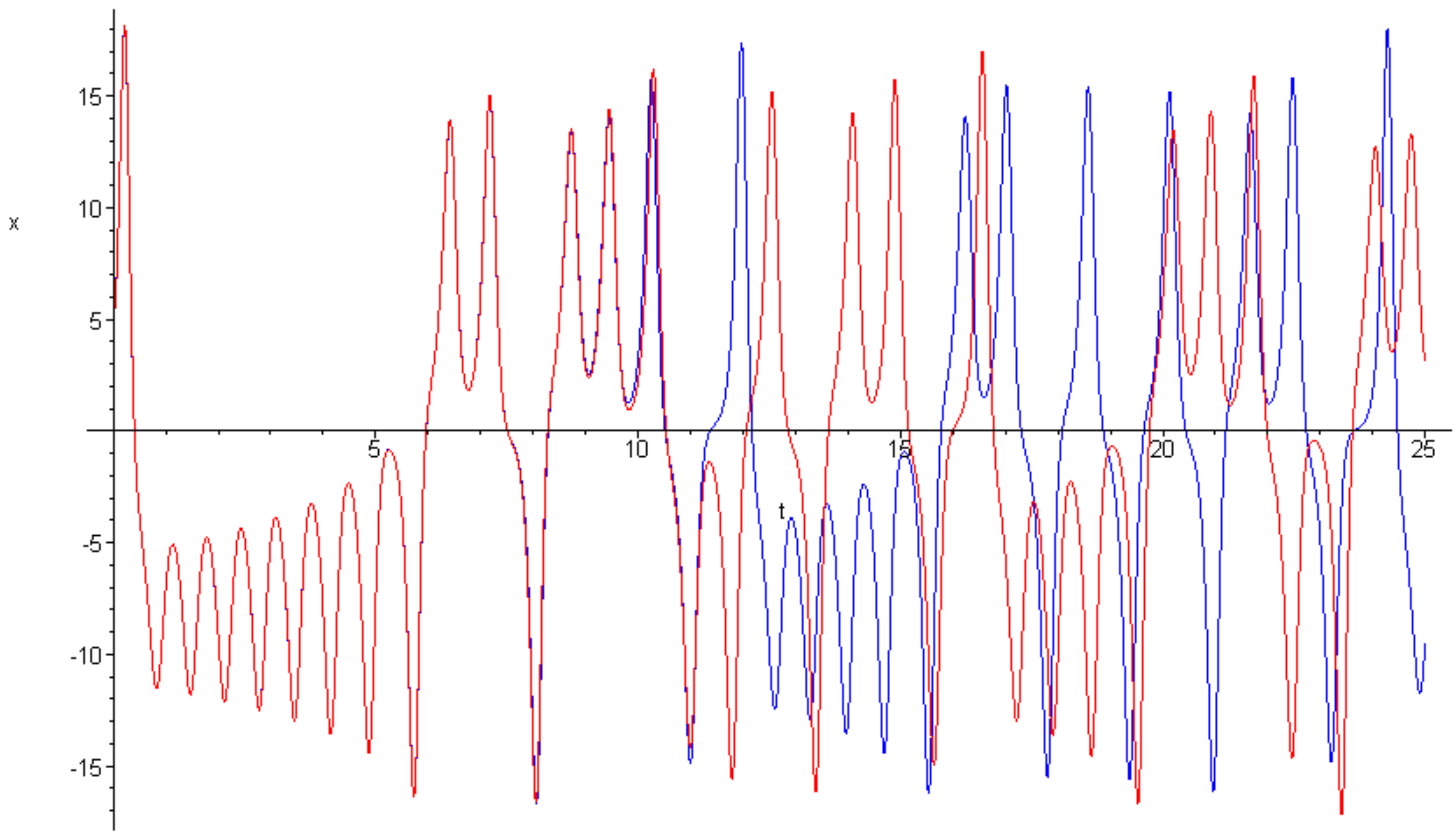
Lorenz Equations

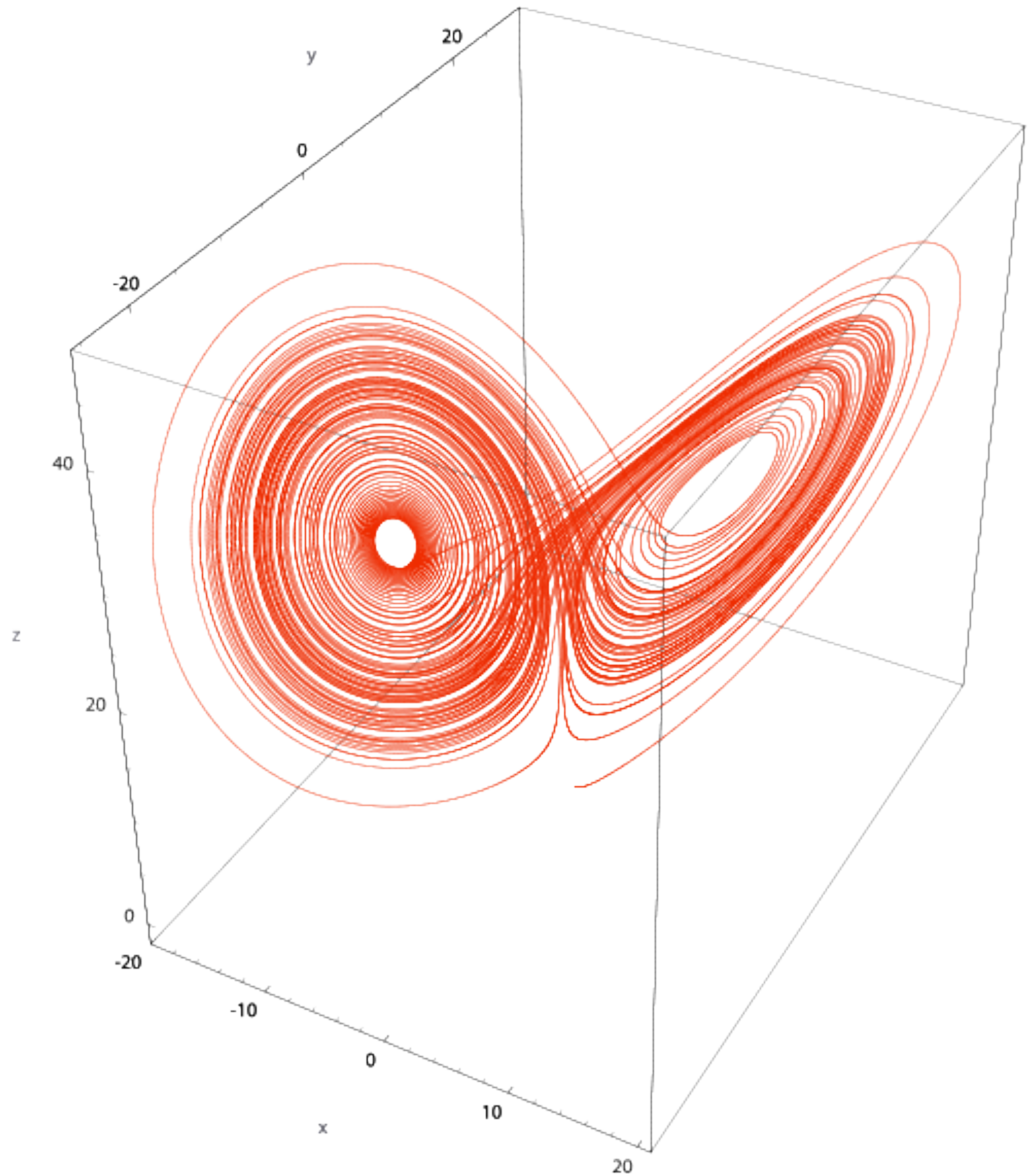
$$\frac{dx}{dt} = -\sigma x + \sigma y$$

$$\frac{dy}{dt} = rx - y - xz$$

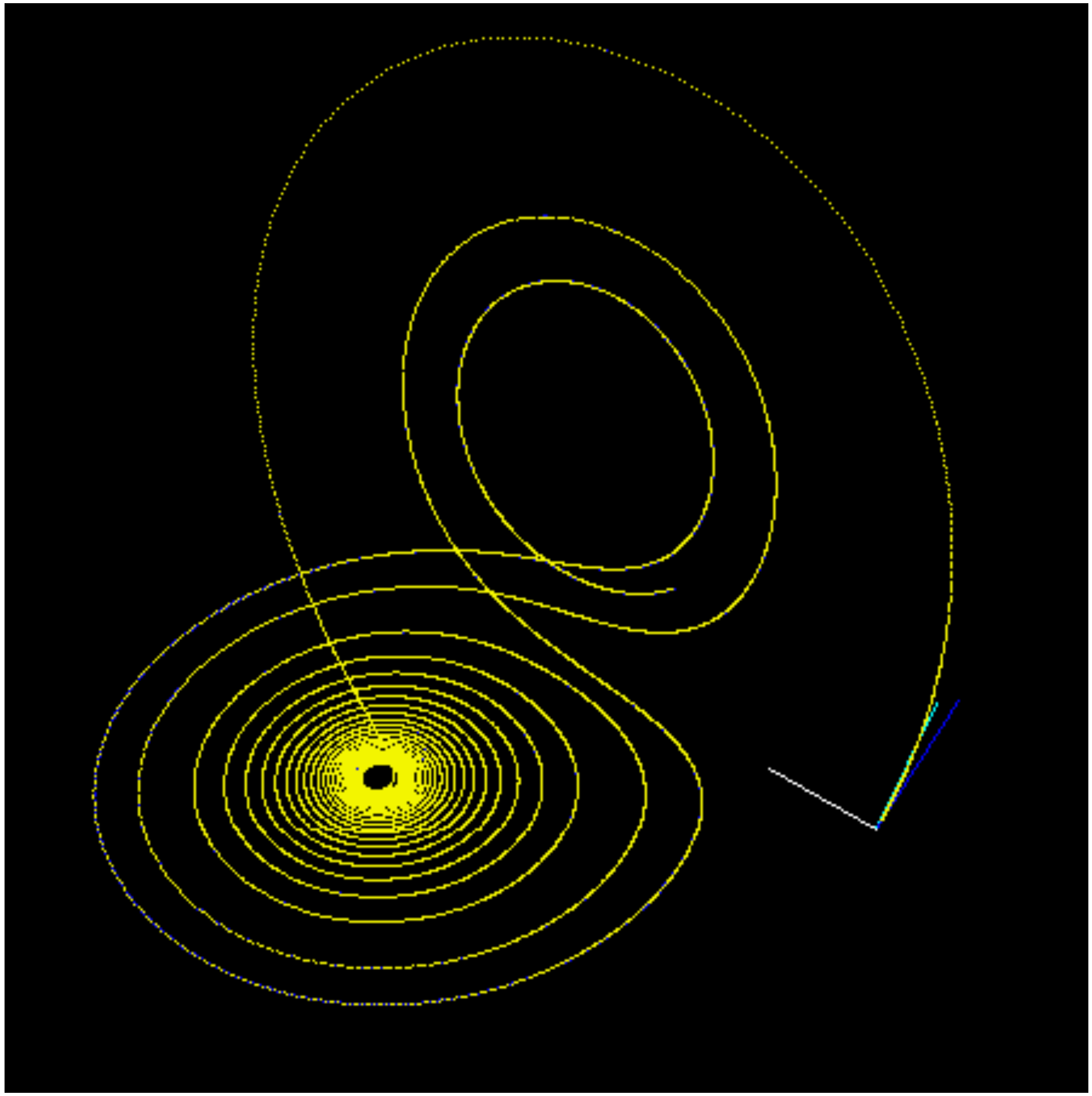
$$\frac{dz}{dt} = -bz + xy$$

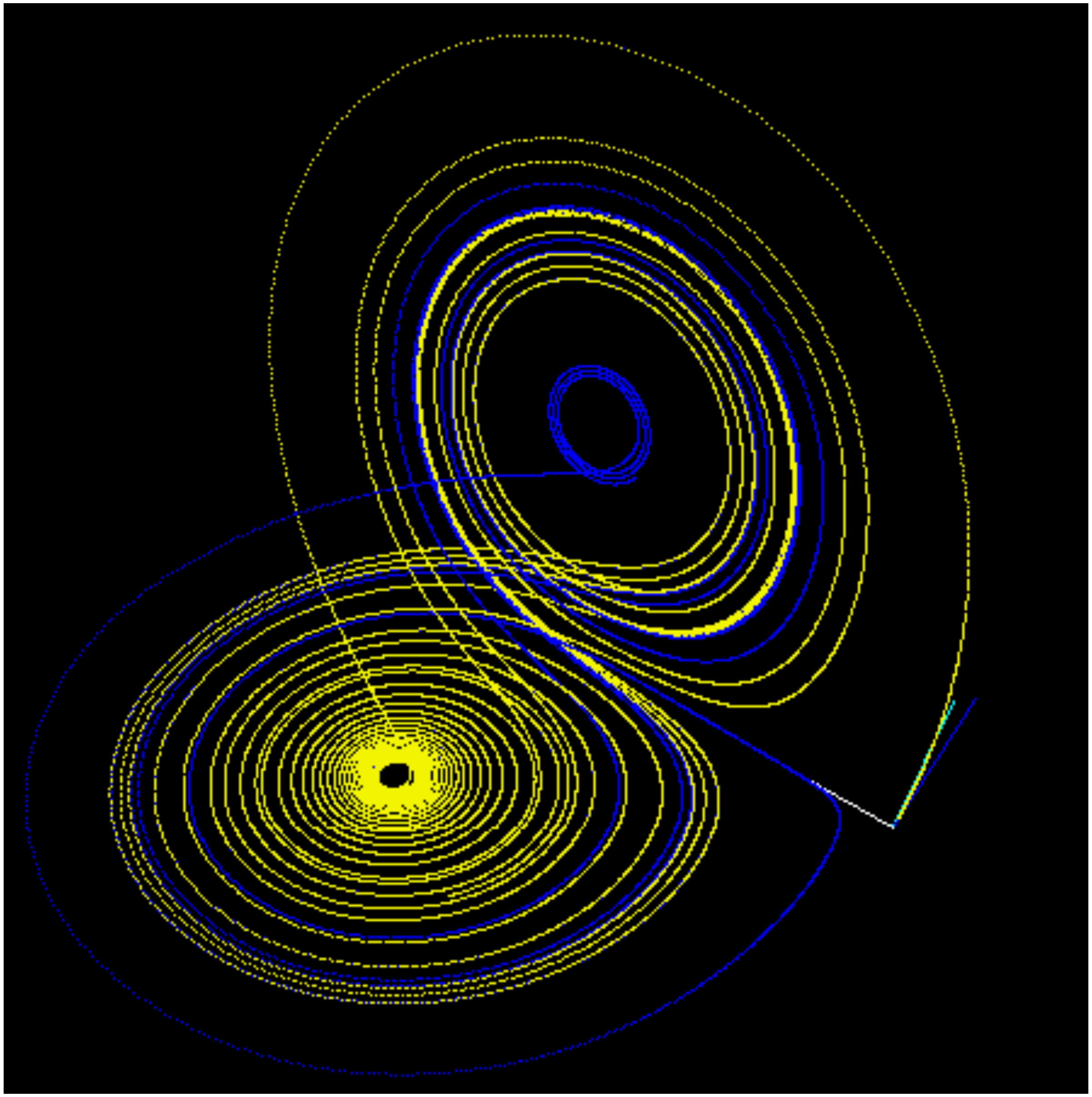
x over time for 2 initial conditions

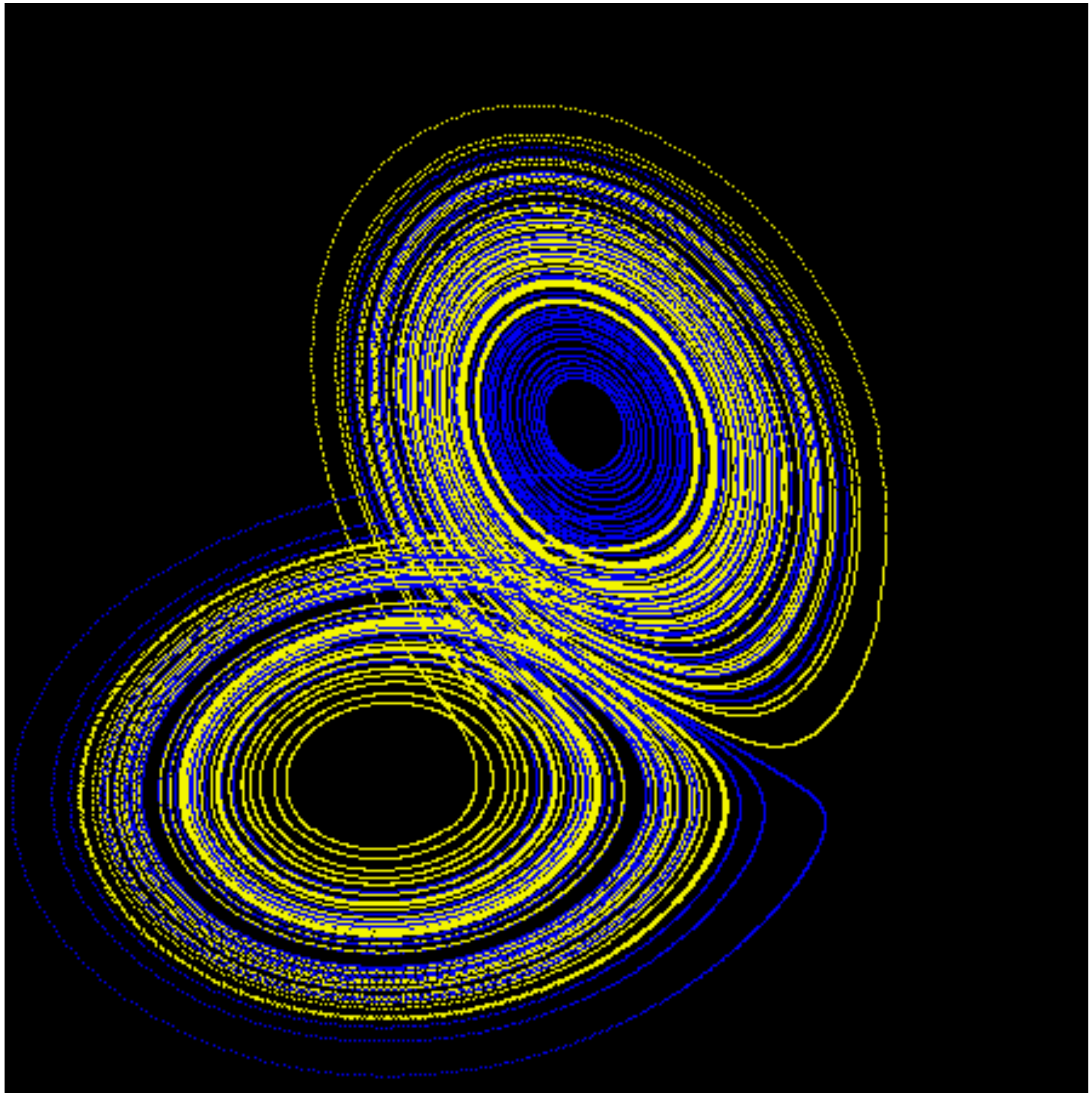




y
20
0
-20
40
z
20
0
-20
-10
0
10
20
x







Logistic Map

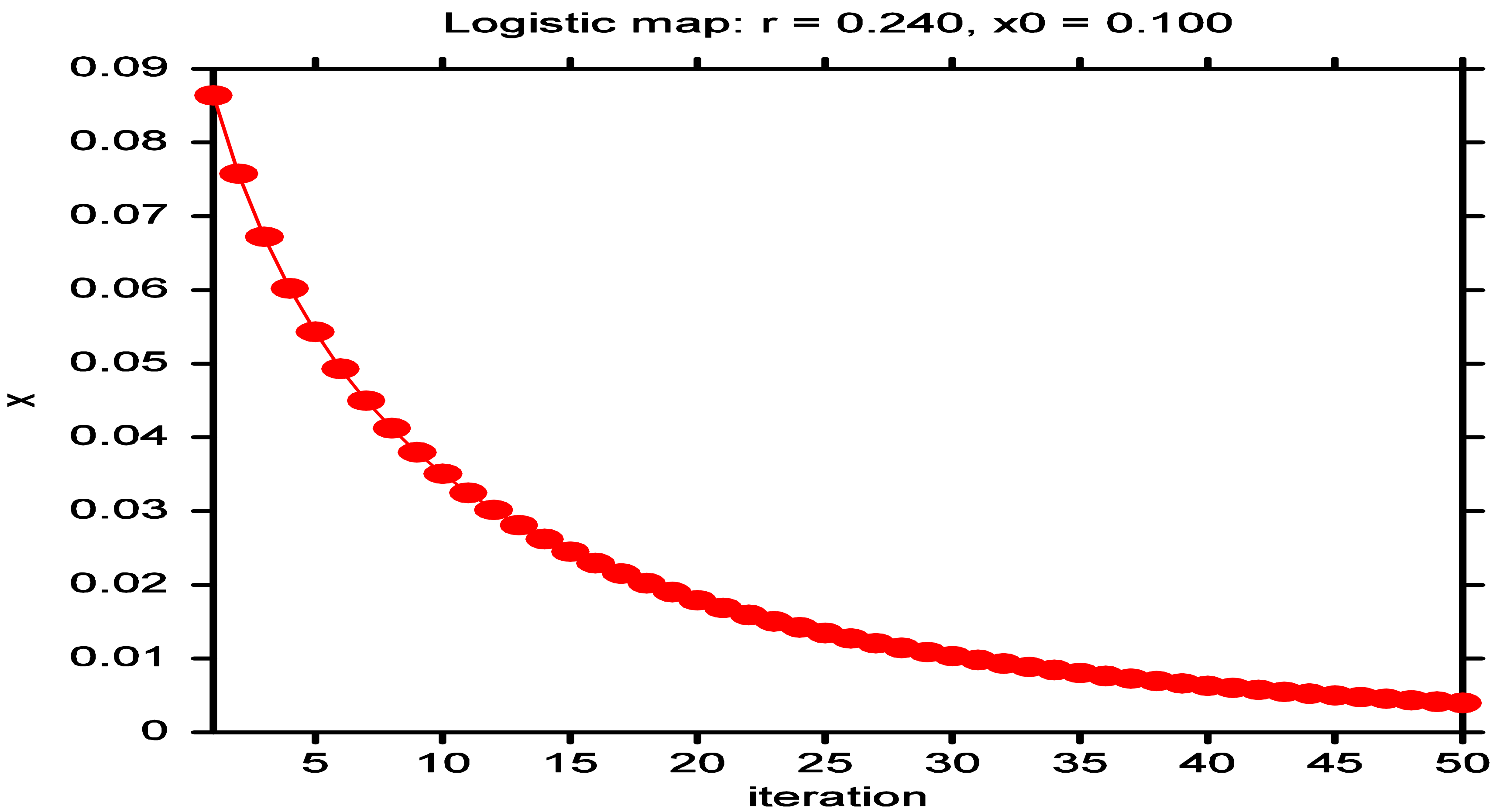
- Verhulst equation:

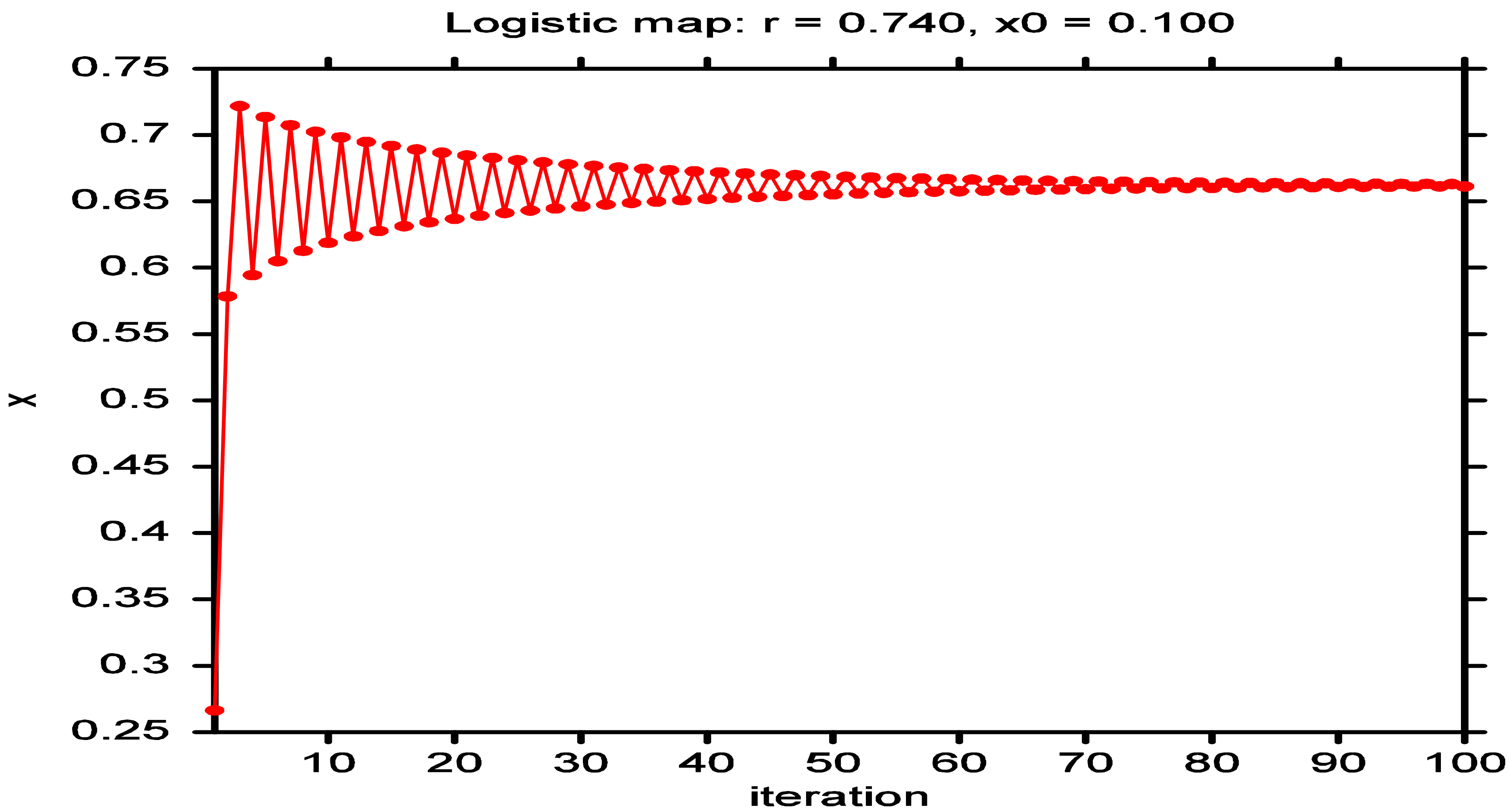
$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$$

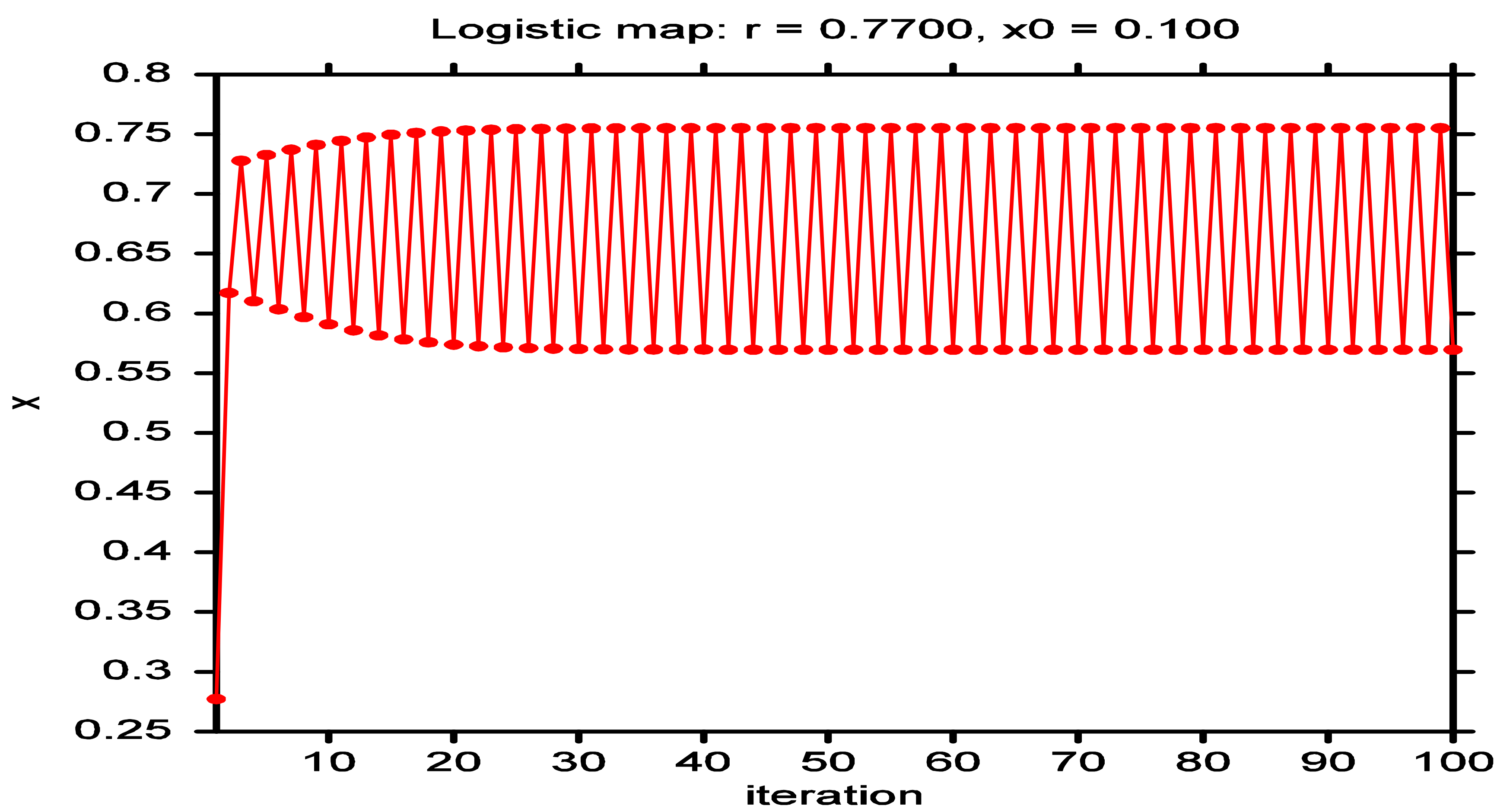
- Logistic map:

$$x_{n+1} = 4rx_n(1 - x_n)$$

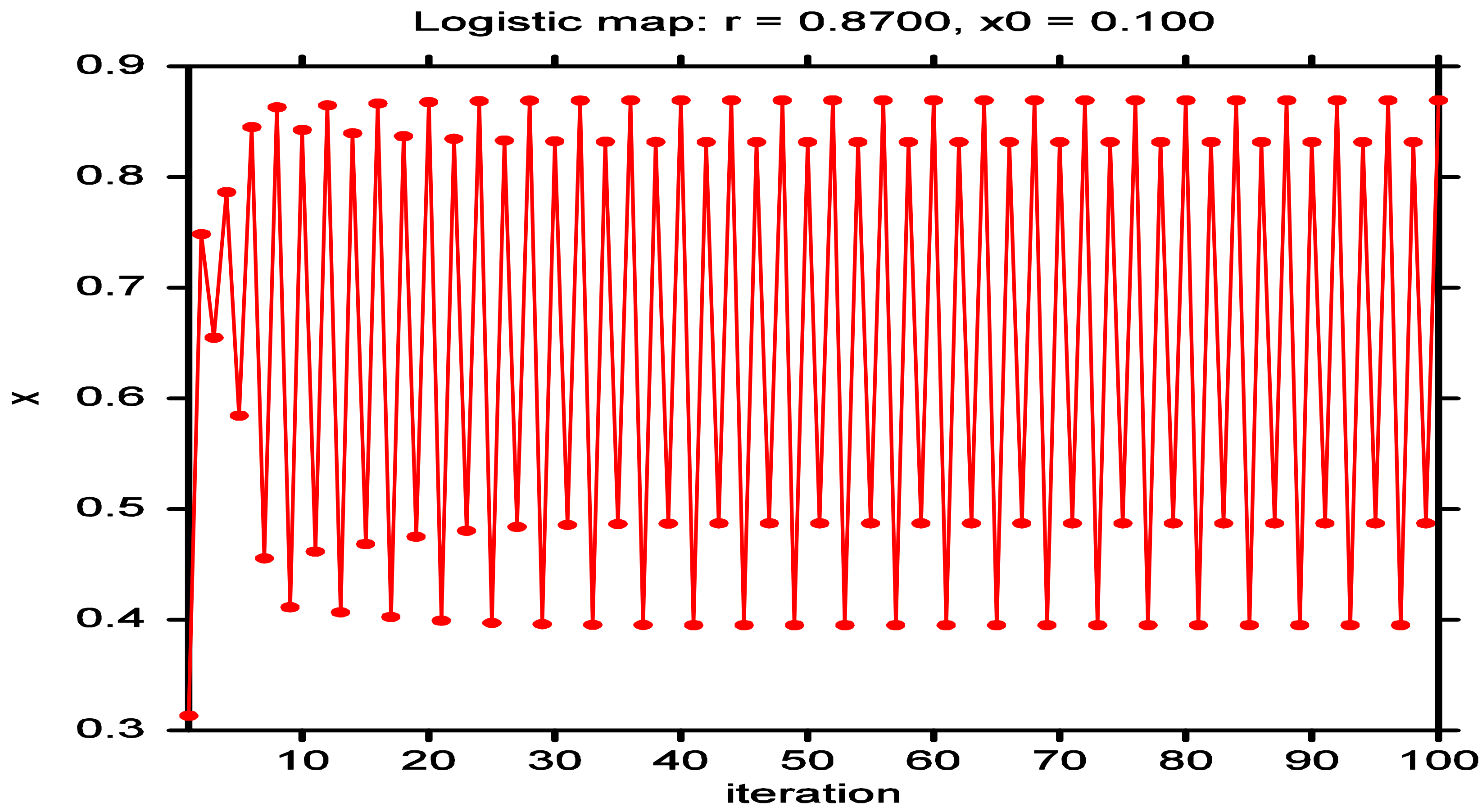
Maps [0,1] → [0,1]

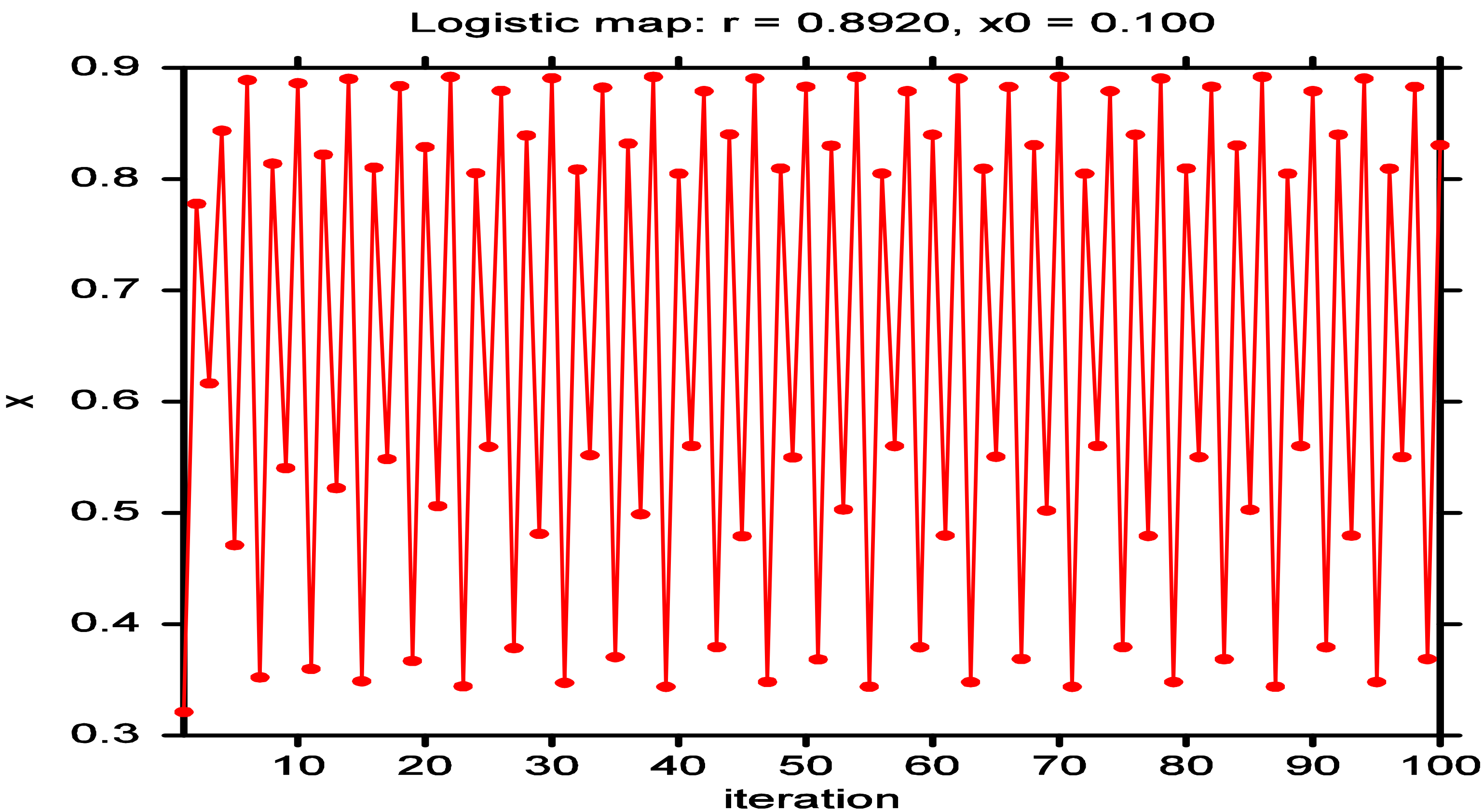

Logistic map: r = 0.240, x0 = 0.100
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
x
5
10
15
20
25
30
35
40
45
50
iteration

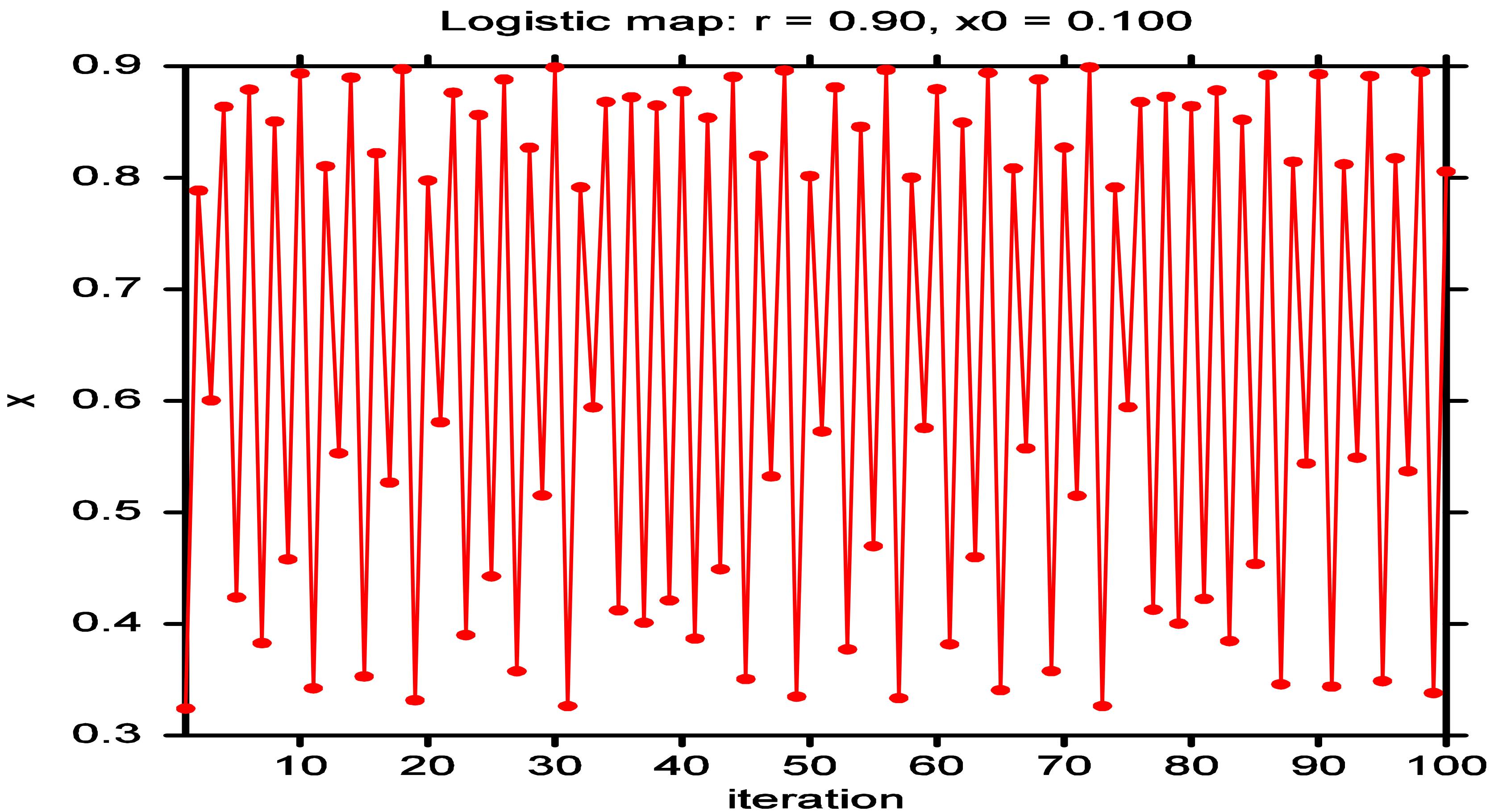

Logistic map: r = 0.740, x0 = 0.100
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
x
10
20
30
40
50
60
70
80
90
100
iteration

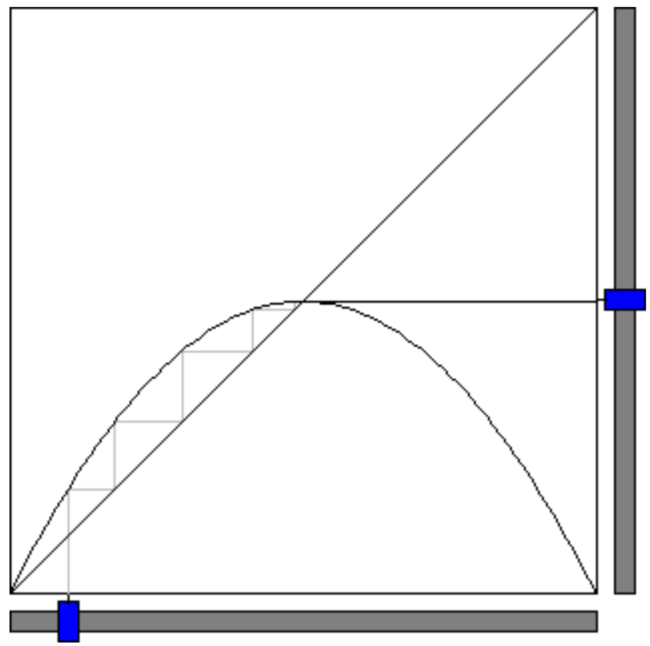


Logistic map: r = 0.7700, x0 = 0.100
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
x
10
20
30
40
50
60
70
80
90
100
iteration

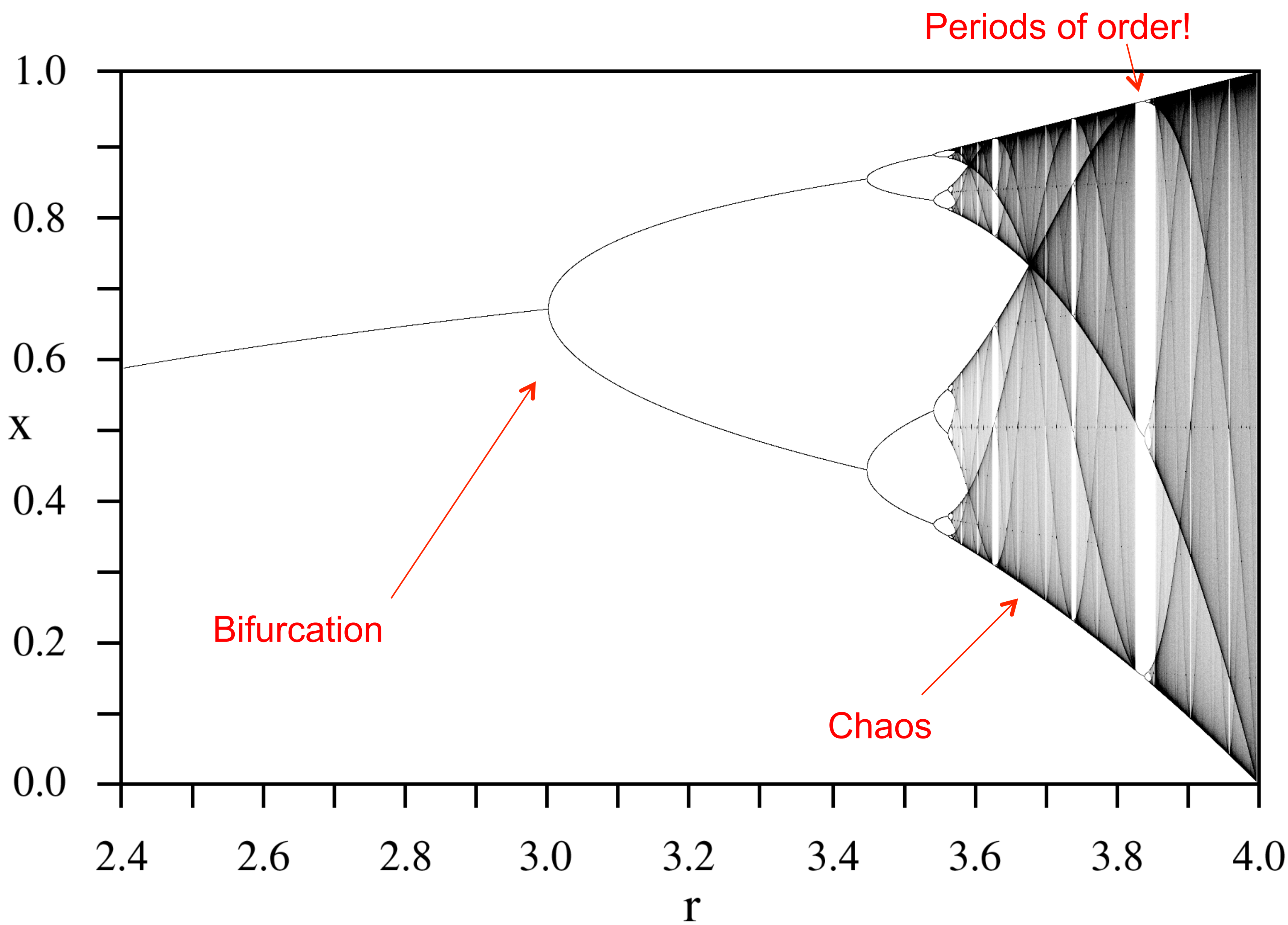

Logistic map: r = 0.8700, x0 = 0.100
x
0.9
0.8
0.7
0.6
0.5
0.4
0.3
10
20
30
40
50
60
70
80
90
100
iteration


Logistic map: r = 0.8920, x0 = 0.100
0.9
0.8
0.7
0.6
0.5
0.4
0.3
x
10
20
30
40
50
60
70
80
90
100
iteration

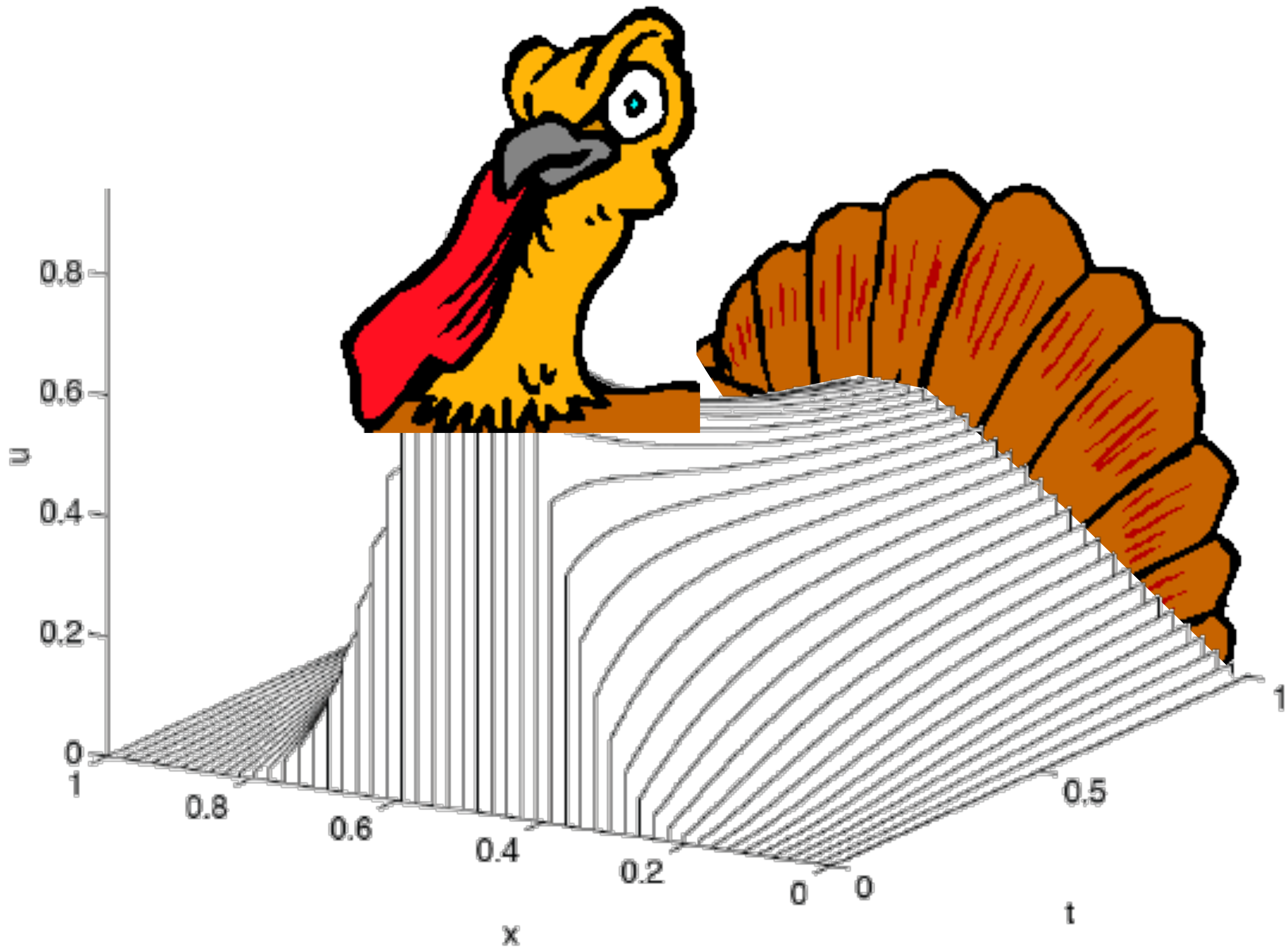

Logistic map: r = 0.90, x0 = 0.100
0.9
0.8
0.7
0.6
0.5
0.4
0.3
x
10
20
30
40
50
60
70
80
90
100
iteration



Iterated Logistic Map Demo:
http://ibiblio.org/e-notes/MSet/Logistic.htm



Bifurcation diagram



u
0.8
0.6
0.4
0.2
0
1
0.8
0.6
0.4
0.2
0
x
0
0.5
1
t