Lecture 8

The Chaotic Motion of Dynamical Systems

Hai-Qing Lin

Beijing Computational Science Research Center

This PowerPoint Notes Is Based on the Textbook '***An Introduction to Computer Simulation Methods : Applications to Physical Systems***', 2nd Edition, Harvey Gould and Jan Tobochnik, Addison-Wesley(1996); "A First Course in Computational Physics"; "Numerical Recipes"; "Elementary Numerical Analysis"; "Computational Methods in Physics and Engineering".

Introduction

- Nature is full of ***nonlinear*** phenomena: weather, turbulent motion, soliton, human body, even Newtonian mechanics. Very very few are linear.
- Chaos is a good example of nonlinear phenomena which has been studied for more than 100 years.
- Computer simulation is a popular tool for making empirical observations of nonlinear phenomena.

Objectives and Requirements

- Use a simple model, the one dimensional logistic, to study some common features of ***nonlinear*** phenomena.
- Master basic concepts and frequently applied techniques in analyzing nonlinear phenomena.

A Simple One-Dimensional Map

- **Example of population growth**: (assuming linear)

  $$P_{n+1} = aP_n$$

  where P_n is the population in generation n and a is a constant. This model is obviously inadequate, 0 or ∞.

- It is more realistic for a model to have the population bounded by the finite carrying capacity of its environment:

  $$P_{n+1} = P_n(a - bP_n),$$

- a nonlinear model where the linear term represents the natural growth of the population, while the quadratic term represents a reduction of this natural growth caused.

A Simple One-Dimensional Map

- Let $P_n = (a/b)\, x_n$ and $a = 4r$, we obtain

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

- This is the one-dimensional ***logistic map*** to be studied with the following properties:
  - its dynamics are determined by a single control parameter r;
  - the function f transfers any point on [0,1] into [0,1];
  - to avoid unphysical features, we impose that

    $$0 \le x_n \le 1, \qquad 0 \le r \le 1,$$
  - this dynamical system is deterministic, the sequence (***trajectory/orbit***) starts from x_0 (***seed***), x_1, x_2, $\cdots$.

The following program computes the trajectory for the logistic map:

```
PROGRAM LOGISTIC
IMPLICIT NONE
integer :: ntrial, nplot, ntimes
integer :: i, j
real :: r, x(99)
write(0,*) 'Enter Parameter r'
read(*,*)  r
write(0,*) 'Enter Ntrial, Nplot'
read(*,*) ntrial, nplot
write(0,*) 'Enter Ntimes'
read(*,*) ntimes
write(0,*) 'Enter x_0 (Total of Ntimes)'
read(*,*) (x(i), i=1, ntimes)
if(ntrial >= 1) then      !Toward Eqm.
   do j = 1, ntrial
       call iterate(r,x,ntimes)
   end do
end if
do j = 1, nplot
     call iterate(r,x,ntimes)
     write(6,1002) (x(i), i=1,ntimes)
end do
1002 format(8f9.6)
stop
END PROGRAM LOGISTIC
```

```
SUBROUTINE  ITERATE(r,x,ntimes)
IMPLICIT NONE
integer :: ntimes,i
real      :: r,parameter,x(ntimes)
parameter = 4.0*r
do i = 1, ntimes
     x(i) = parameter*x(i)*(1.0-x(i))          ! Function form
end do
END SUBROUTINE ITERATE
```

Real **16**

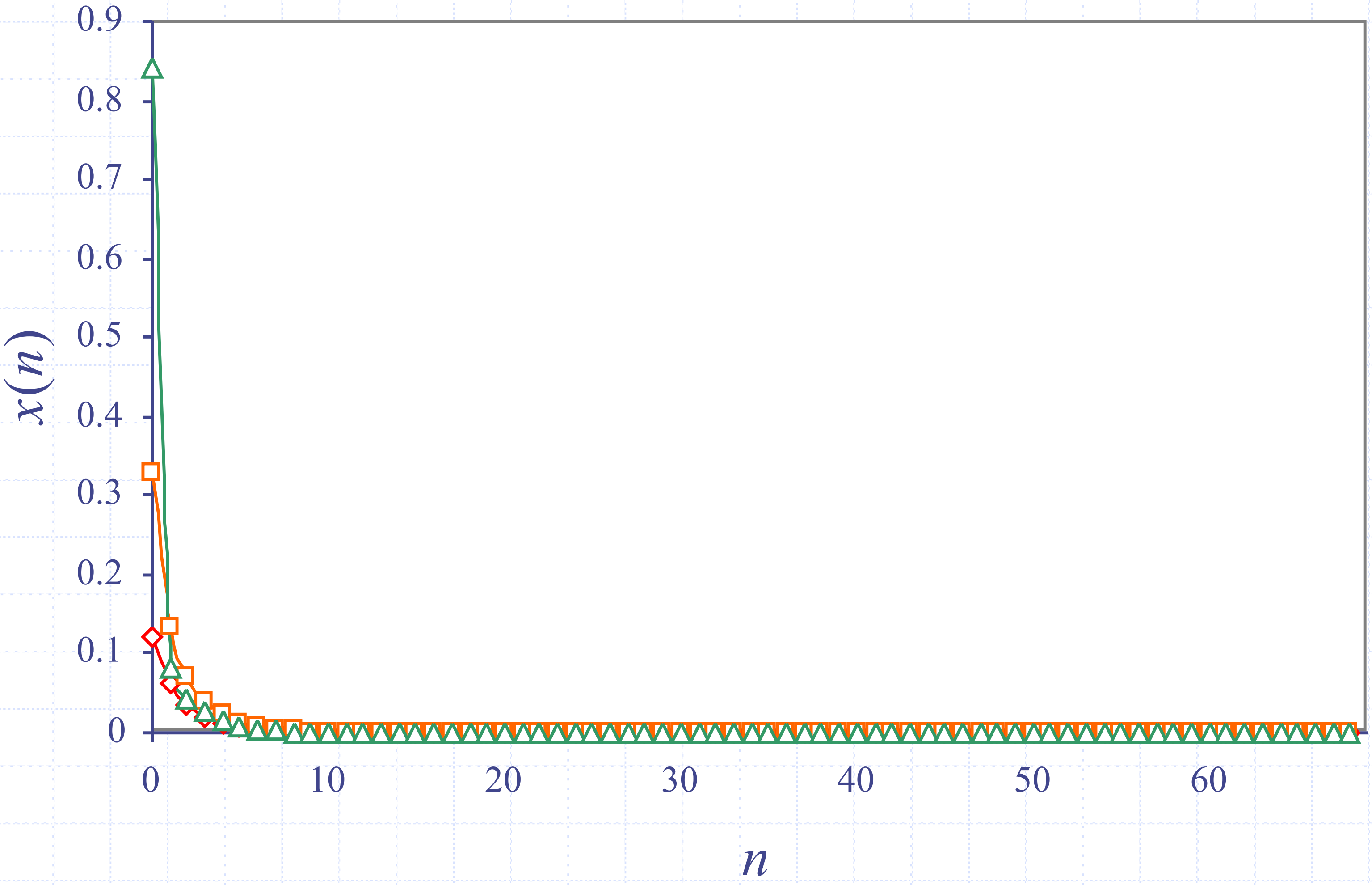
$$x_{n+1} = f(x_n) = ax_n(1 - x_n) = 4rx_n(1 - x_n)$$

r = 0.150

n	Seq.: 1	2	3
0	**0.120000**	**0.330000**	**0.840000**
1	0.063360000000000	0.132660000000000	0.080640000000000
2	0.035607306240000	0.069036794640000	0.044482314240000
53	0.000000000000158	0.000000000000284	0.000000000000194
54	0.000000000000095	0.000000000000170	0.000000000000116
55	0.000000000000057	0.000000000000102	0.000000000000070
56	0.000000000000034	0.000000000000061	0.000000000000042
57	0.000000000000021	0.000000000000037	0.000000000000025
58	0.000000000000012	0.000000000000022	0.000000000000015
59	0.000000000000007	0.000000000000013	0.000000000000009
60	0.000000000000004	0.000000000000008	0.000000000000005
61	0.000000000000003	0.000000000000005	0.000000000000003
62	0.000000000000002	0.000000000000003	0.000000000000002
63	0.000000000000001	0.000000000000002	0.000000000000001
64	0.000000000000001	0.000000000000001	0.000000000000001
65	0.000000000000000	0.000000000000001	0.000000000000000
67	0.000000000000000	0.000000000000000	0.000000000000000
68	0.000000000000000	0.000000000000000	0.000000000000000

Stable ($r < 3/4$) - x_n vs iteration n

Initial values $x_0 = 0.12$, 0.33, 0.84, $x_{n\to\infty} = 0.00$



Real **16**

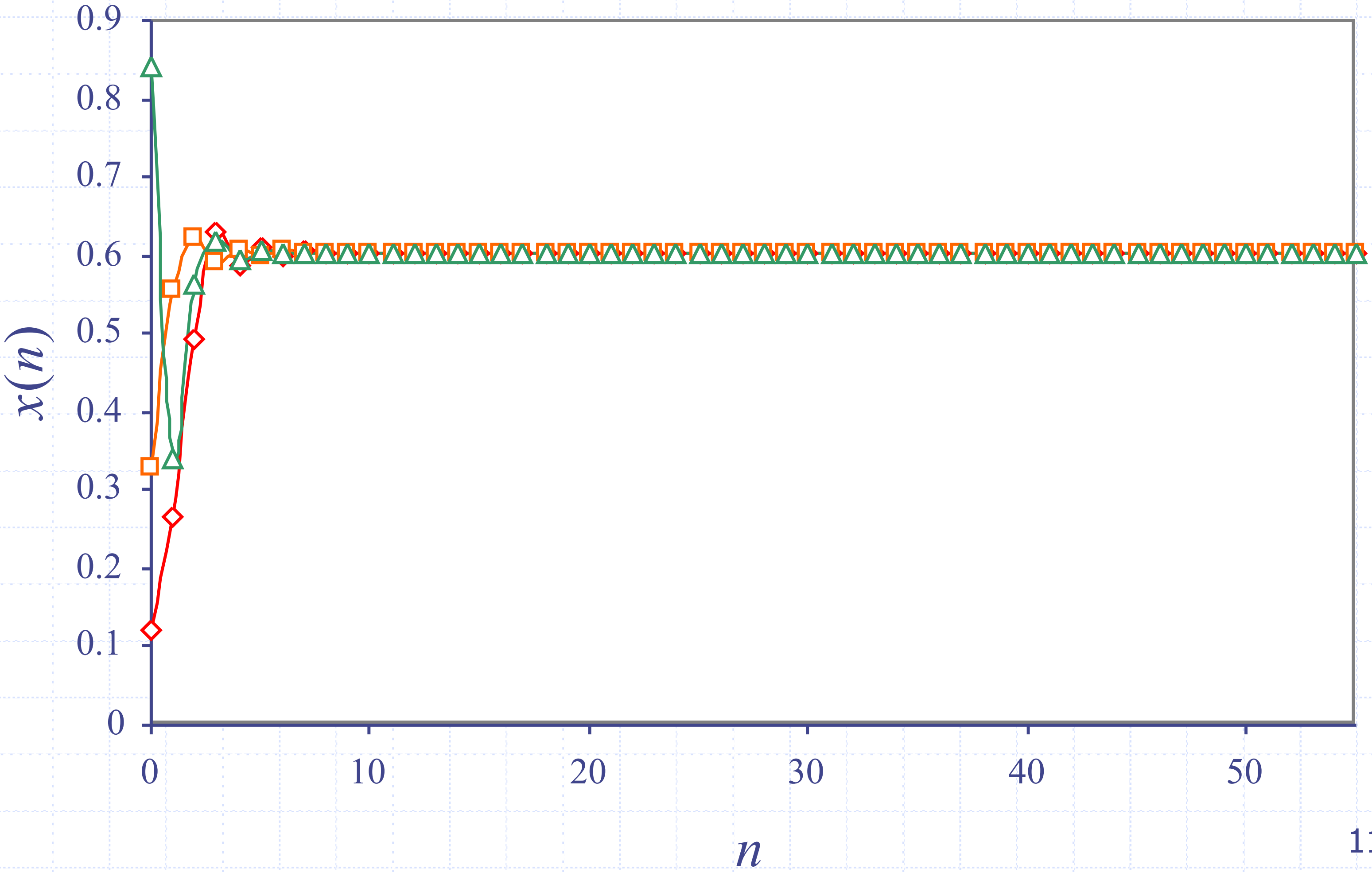
r = 0.630

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

n	Seq.: 1	2	3
0	0.120000	0.330000	0.840000
1	0.266112000000000	0.557172000000000	0.338688000000000
2	0.492146936709120	0.621763033288320	0.564425665413120
38	0.603174603171334	0.603174603175770	0.603174603172638
39	0.603174603176303	0.603174603173997	0.603174603175625
40	0.603174603173719	0.603174603174919	0.603174603174072
41	0.603174603175063	0.603174603174439	0.603174603174880
42	0.603174603174364	0.603174603174688	0.603174603174459
43	0.603174603174727	0.603174603174559	0.603174603174678
44	0.603174603174539	0.603174603174626	0.603174603174564
45	0.603174603174637	0.603174603174591	0.603174603174623
46	0.603174603174586	0.603174603174609	0.603174603174593
47	0.603174603174612	0.603174603174600	0.603174603174609
48	0.603174603174598	0.603174603174605	0.603174603174600
49	0.603174603174606	0.603174603174602	0.603174603174605
50	0.603174603174602	0.603174603174604	0.603174603174602
51	0.603174603174604	0.603174603174603	0.603174603174604
52	0.603174603174603	0.603174603174603	0.603174603174603
53	0.603174603174603	0.603174603174603	0.603174603174603
54	0.603174603174603	0.603174603174603	0.603174603174603

Stable (r < 3/4) - x_n vs iteration n

Initial values $x_0 = 0.12, 0.33, 0.84$, $x_{n\to\infty} = 0.60$



Fixed Point and Its Stability

Fixed point: $x^* = f(x^*)$ for *large enough* ***n***

- From $x_{n+1} = 4rx_n(1 - x_n)$, we obtain
  $x^* = 4r\,x^*(1 - x^*) \;\Rightarrow\; x^* = 0$ and $x^* = 1 - 1/(4\,r)$
- There exists only one fixed point $x^* = 0$ for $r <$ 1/4.

Stable fixed point:

- The iterated values of x converges to x^* ***independently*** of the value of x_0, the initial seed.
- A stable fixed point is also called ***an attractor of period 1***.

Stability of x^*:

- Start from $x_n = x^* + \varepsilon_n$ and $x_{n+1} = x^* + \varepsilon_{n+1}$

$$x^* + \varepsilon_{n+1} = x_{n+1} = f(x^* + \varepsilon_n)$$
$$= f(x^*) + \varepsilon_n f'(x^*)$$
$$= x^* + \varepsilon_n f'(x^*)$$
$$\Rightarrow \varepsilon_{n+1}/\varepsilon_n = f'(x^*)$$

- $\therefore$ The local stability criteria for fixed point x^* are

Stable	$\lvert f'(x^*)\rvert < 1$
Marginally stable	$\lvert f'(x^*)\rvert = 1$, consider $f''(x^*)$
Unstable	$\lvert f'(x^*)\rvert > 1$

- $x^* = ½$ $(f'(x^*) = 0)$ is a ***superstable*** fixed point.

Stability of the 1D logistic map:

- $f'(x) = 4r - 8rx$

  $f'(x^* = 0) = 4r$

  $f'(x^* = 1 - 1/4r) = 2 - 4r.$

- Thus

  $x^* = 0$ is stable if $r < 1/4$.

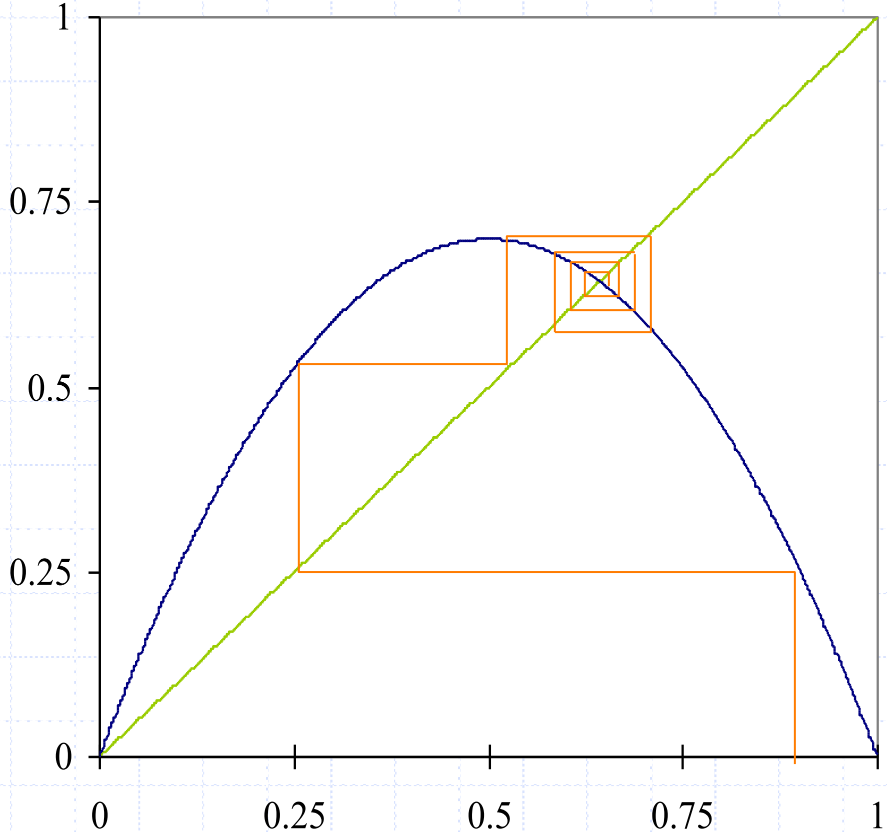
  $x^* = 1 - 1/4r$ is stable if $1/4 < r < 3/4$.

Graphic Solution

A graphic method for iterating $x_{n+1} = 4rx_n(1 - x_n)$:

We draw $y = x$ and $y = f(x)$ (2 intersections) and then start from any initial values x_0, then continued iterations will converge to one of them if $r < 3/4$.

The iteration of the logistic map with $r = 0.7$ and $x_0 = 0.9$



Program of graphic solution: **graph_sol** (p.138-139).

Variation of Parameter *r*

- For fixed point of period 1, $x^* = f(x^*)$, we have

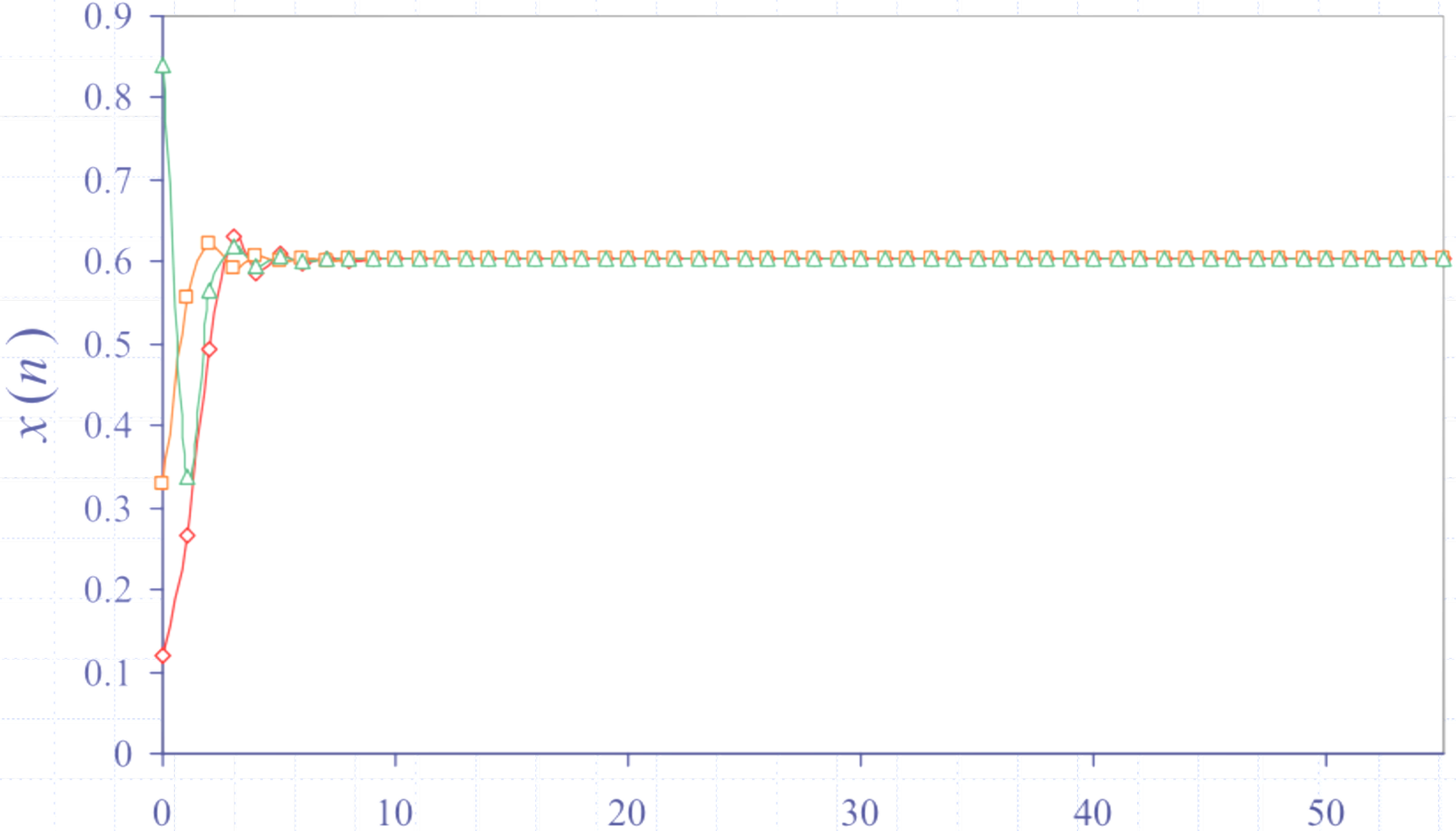
  $x^* = 0$ is stable if $r < 1/4$.

  $x^* = 1 - 1/4r$ is stable if $1/4 < r < 3/4$.

- What will happen **if r > 3/4 ?**

Period 2 ($3/4 < r < 0.8624$) – x_n vs iteration n

Initial values x_0 = 0.12, 0.33, 0.84, $x_{n\to\infty}$ = **0.49/0.82**



Real **16**

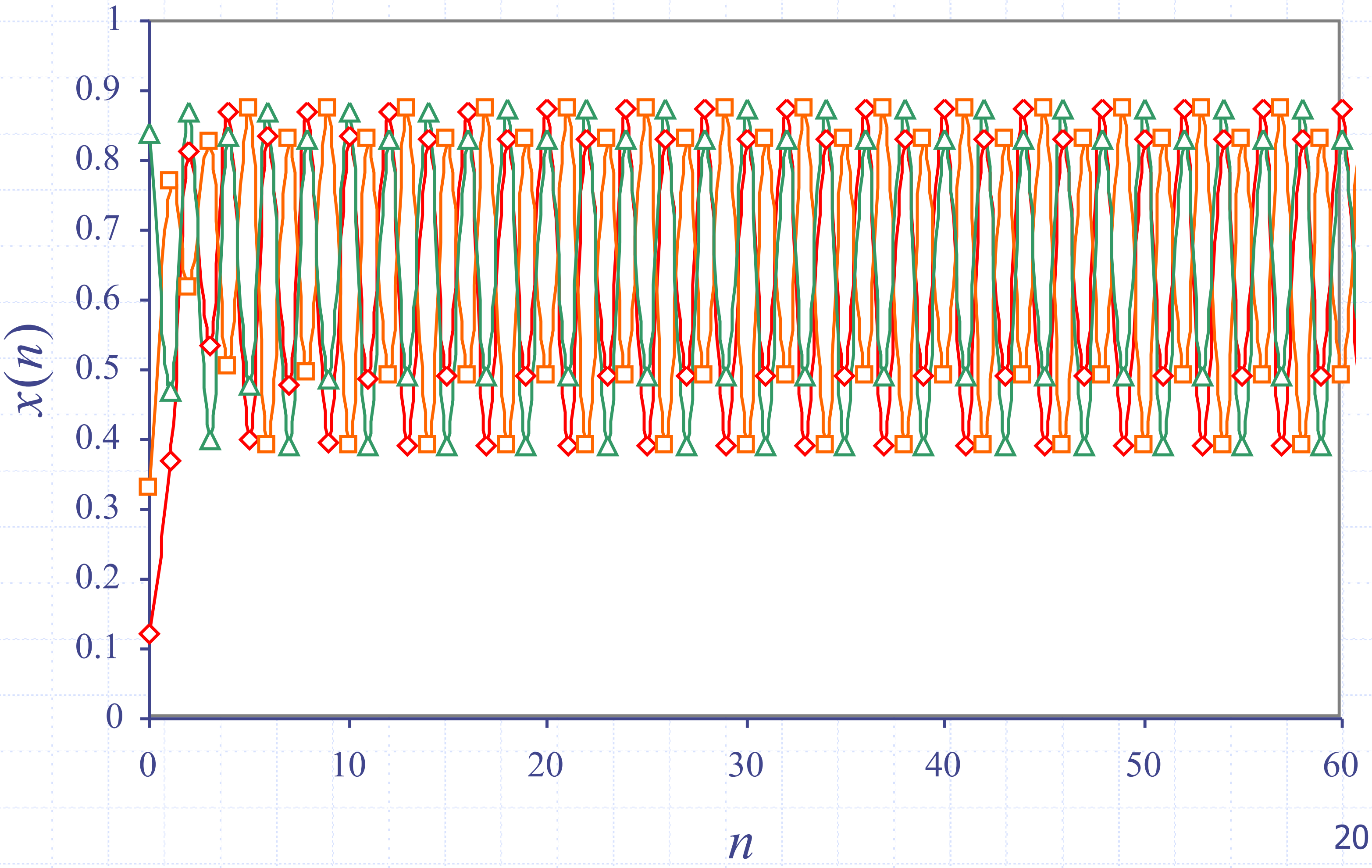
$$x_{n+1} = f(x_n) = ax_n(1 - x_n) = 4rx_n(1 - x_n)$$

r = 0.820

n	Seq.: 1	2	3
0	**0.120000**	**0.330000**	**0.840000**
1	0.346368000000000	0.725208000000000	0.440832000000000
2	0.742582844129280	0.653642850094080	0.808517204705280
51	0.485561802641415	0.819316246139073	0.485561802641415
52	0.819316246139073	0.485561802641415	0.819316246139073
53	0.485561802641415	0.819316246139073	0.485561802641415
54	0.819316246139073	0.485561802641415	0.819316246139073
55	0.485561802641415	0.819316246139073	0.485561802641415
56	0.819316246139073	0.485561802641415	0.819316246139073
57	0.485561802641415	0.819316246139073	0.485561802641415
58	0.819316246139073	0.485561802641415	0.819316246139073
59	0.485561802641415	0.819316246139073	0.485561802641415
60	0.819316246139073	0.485561802641415	0.819316246139073
61	0.485561802641415	0.819316246139073	0.485561802641415
62	0.819316246139073	0.485561802641415	0.819316246139073
63	0.485561802641415	0.819316246139073	0.485561802641415

Period 4 (0.8624 < r < 0.8860) - x_n vs iteration n

Initial values $x_0 = 0.12$, 0.33, 0.84, $\boldsymbol{x_{n\to\infty}}$ **= 0.39/0.49/0.83/0.87**



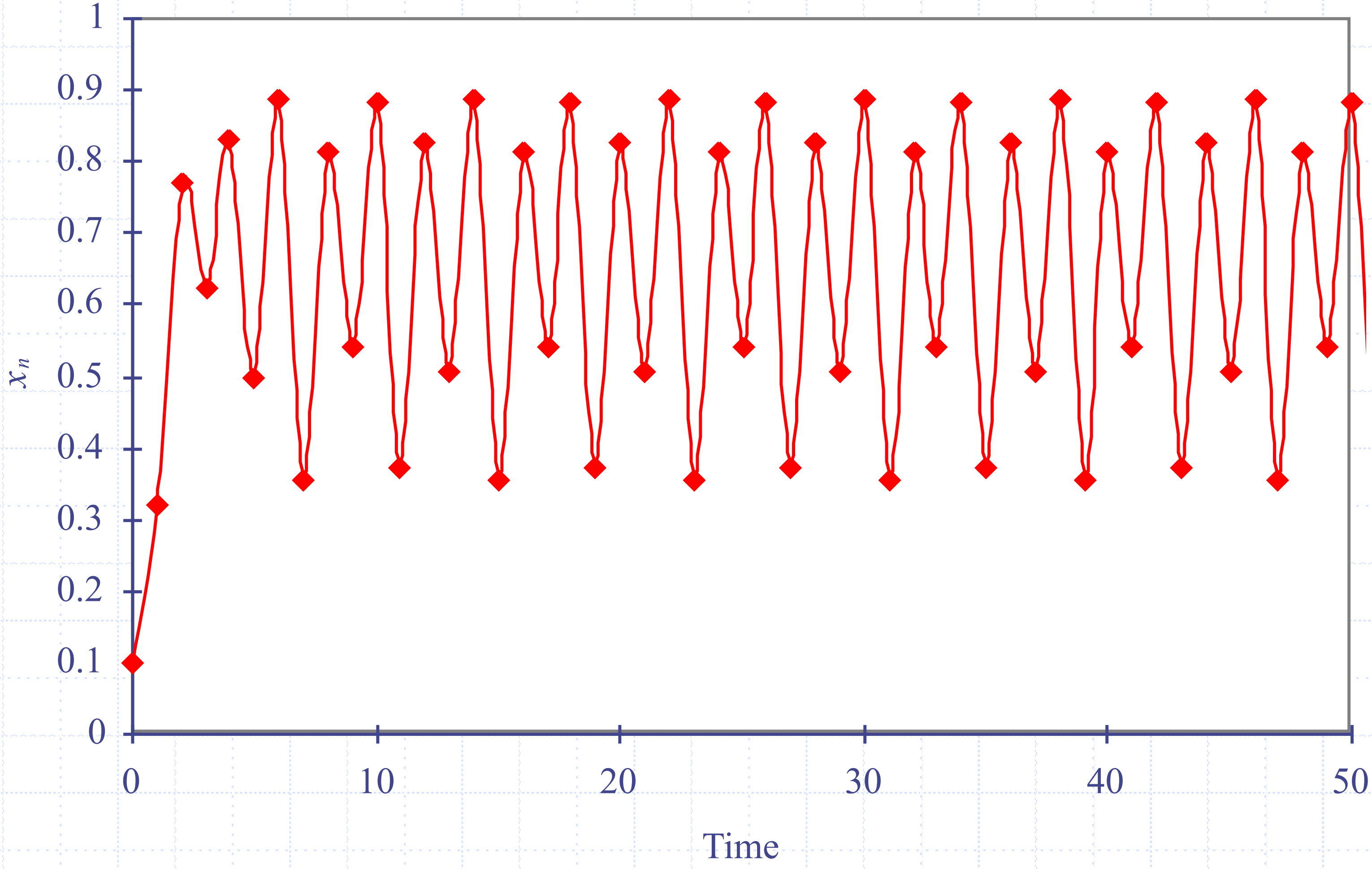
Real 16

r = 0.87200

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

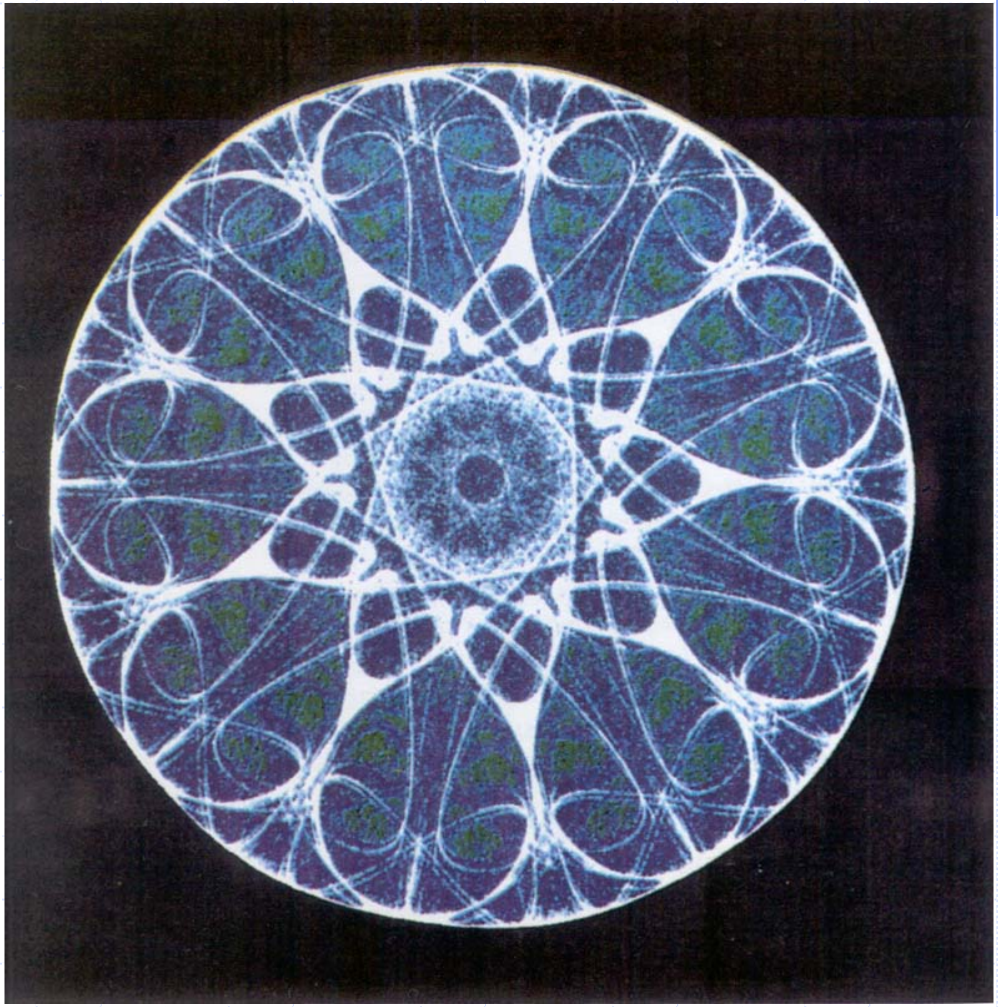
n	Seq.: 1	2	3
0	0.120000	0.330000	0.840000
1	0.368332800000000	0.771196800000000	0.468787200000000
2	0.811531154573230	0.615465607296123	0.868601854773166
208	0.871832346529560	0.493067050516292	0.829604359319254
209	0.389751582787233	0.871832346529560	0.493067050516292
210	0.829604359319254	0.389751582787233	0.871832346529560
211	0.493067050516292	0.829604359319254	0.389751582787233
212	0.871832346529560	0.493067050516292	0.829604359319254
213	0.389751582787233	0.871832346529560	0.493067050516292
214	0.829604359319254	0.389751582787233	0.871832346529560
215	0.493067050516292	0.829604359319254	0.389751582787233
216	**0.871832346529560**	0.493067050516292	0.829604359319254
217	0.389751582787233	0.871832346529560	0.493067050516292
218	0.829604359319254	0.389751582787233	**0.871832346529560**
219	0.493067050516292	0.829604359319254	0.389751582787233
220	**0.871832346529560**	0.493067050516292	0.829604359319254
221	0.389751582787233	**0.871832346529560**	0.493067050516292

Period 8 (0.8860 < r < 0.8911) - x_n vs iteration n



Polynomial Map

The "strange attractor" of a polynomial map of the plane with a 9-fold symmetry. (Figure after P Chossat and M Golubitsky, courtesy of Dr K A Cliffe, Harwell Laboratories, U.K.A.E.A.)



Multiple Fixed Points

- It is possible that, after some iterations, we get

$$x_{n+\mathbf{p}} = x_n, \quad x_{n+\mathbf{p}+1} = x_{n+1}, \quad etc.$$

then the trajectory has more than one fixed point.

- Denote $f^{(p)}(x)$ as the pth iterate of $f(x)$, e.g., $p = 2$,

$$f^{(2)}(x) = f(f(x)) = 16r^2x(1-x)\,[1-4rx(1-x)]$$

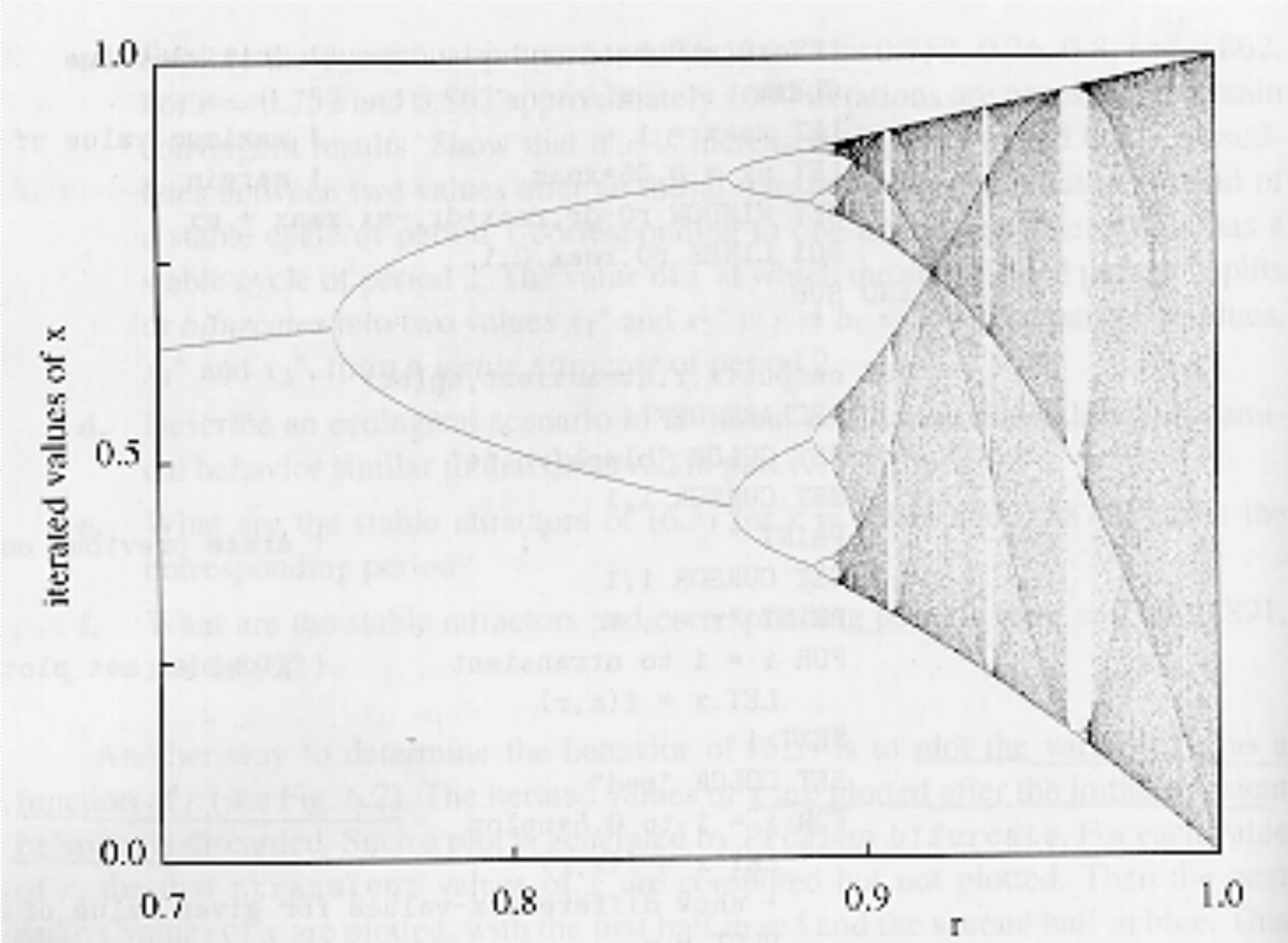
- Then (multiple) fixed points are determined by

$$\boldsymbol{x^* = f^{(p)}(x^*)}. \quad \text{(not derivative!)}$$

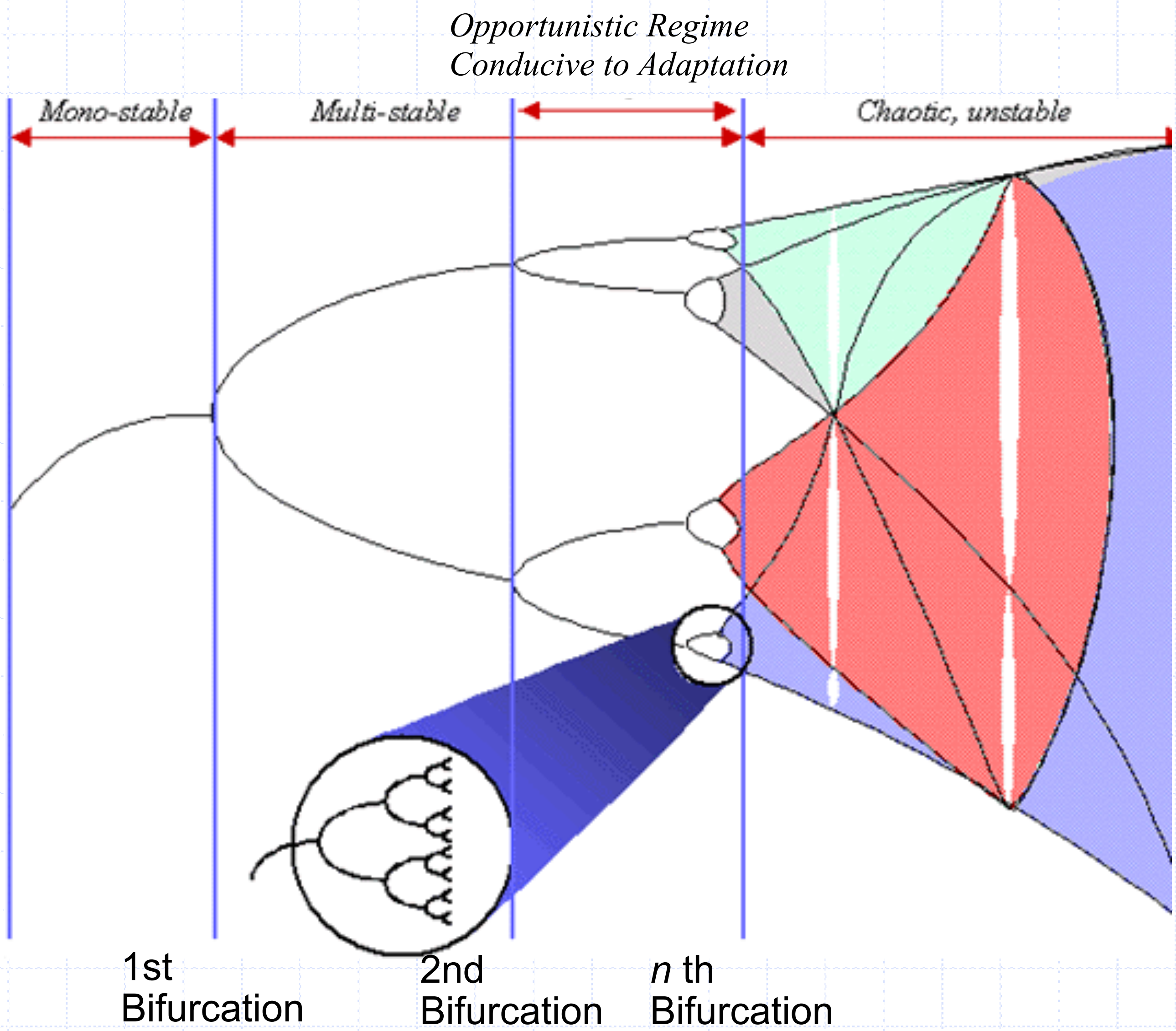
Period-Doubling

- For $r > 3/4$, the fixed point of f becomes unstable and gives birth (***bifurcates***) to a cycle of period **2**.
- As r increase, the fixed points of $f^{(2)}(x)$ become unstable and the cycle of f is period **4**, and so forth.
- Such phenomena is called ***period-doubling***.

Bifurcation Diagram



Bifurcation Diagram www.calresco.org/beckermn/nonlindy.htm ©



→ p.34
→ p.35

Stability of Multiple Fixed Points

- For period two, $f^{(2)}(x) = f(f(x))$ has two fixed points. Take derivative

$$\frac{d}{dx} f^{(2)}(x)|_{x=x_0} = \frac{d}{dx} f(f(x))|_{x=x_0} = f'(f(x_0))f'(x)|_{x=x_0}$$

- Let $x_1 = f(x_0)$, we obtain

$$\frac{d}{dx} f(f(x))|_{x=x_0} = f'(x_1)f'(x_0)$$

$$\frac{d}{dx} f(f(x))|_{x=x_1} = f'(x_1)f'(x_0)$$

- Thus, if x_0 is unstable, so is x_1, for the same value of r, leading to an trajectory of period 4.

Plots the values of x as a function of r. For each r, the 1st *ntransient* (100-1000) values of x are computed but not plotted.

```
!!!!!!!!!!!!!!!   Bifurcation diagram of the logistic map  !!!!!!!!!!!!!!
!!  Compile by the command: f90 bifurcate.f90 graf.a -lX11  !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
PROGRAM bifurcate
IMPLICIT NONE
INTEGER :: nvalues, ntransient, nplot, ir
DOUBLE PRECISION :: x0, r0, rmax, dr, r, xscale, yscale
CHARACTER (4) :: data
  CALL Initial (x0,r0,rmax,nvalues,dr,ntransient,nplot)

  xscale = 400 / (rmax-r0)
  yscale = 400

  CALL INITGRAPH
  CALL LINE (25,425,425,425)
  CALL OUTTEXT(430,425,"r")
  WRITE (data,'(F4.1)') r0
```

```
  CALL OUTTEXT(15,440,data)
  WRITE (data,'(F4.1)') rmax
  CALL OUTTEXT(415,440,data)
  CALL LINE (25,25,25,425)
  CALL OUTTEXT(25,25,"f(x)")
  CALL OUTTEXT(5,35,"1.0")
  CALL OUTTEXT(5,425,"0.0")
  CALL SETCOLOR(5,5,0)

  DO ir = 0, nvalues
    r = r0 + ir*dr
    CALL Output (x0,r0,r,ntransient,nplot,xscale,yscale)
  ENDDO
  CALL Output(x0,r0,rmax,ntransient,nplot,xscale,yscale)

  CALL SETCOLOR(1,5,5)
  CALL OUTTEXT (150,40,"Press any key to exit.")
  CALL GETCH
  CALL CLOSEGRAPH
END PROGRAM bifurcate
```

```
SUBROUTINE Initial (x0,r0,rmax,nvalues,dr,ntransient,nplot)
INTEGER, INTENT(OUT) ::nvalues,ntransient,nplot
DOUBLE PRECISION, INTENT(OUT) :: x0,r0,rmax,dr

!   PRINT *, ' Initial value of control parameter r ='
!   READ  *, r0
!   PRINT *, ' Maximum value of r ='
!   READ  *, rmax                    !! rmax must not be greater than 1
!   PRINT *, ' dr ='
!   READ  *, dr                      !! suggest dr <= 0.01
!   PRINT *, ' no. of iterations not plotted ='
!   READ  *, ntransient
!   PRINT *, ' no. of iterations plotted ='
!   READ  *, nplot

   r0 = 0.7d0
   rmax = 1.d0
   dr = 0.001d0
   ntransient = 100000
   nplot = 250

   nvalues = INT((rmax - r0)/dr)        !! number of r values plotted
   x0 = 0.5d0
END SUBROUTINE Initial
```

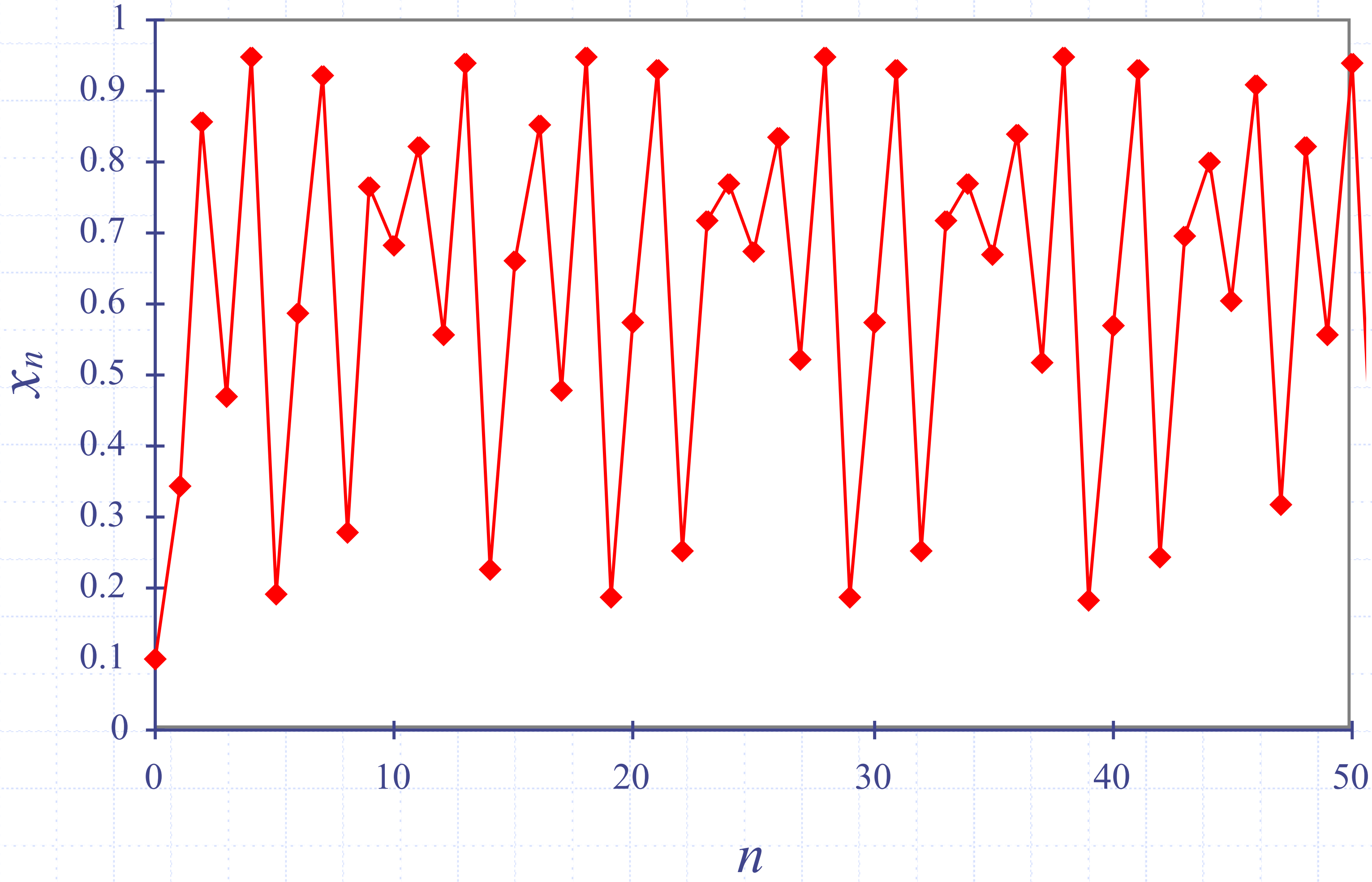
```
SUBROUTINE Output(x0,r0,r,ntransient,nplot,xscale,yscale)
DOUBLE PRECISION, INTENT(IN) :: x0, r0,r, xscale, yscale
INTEGER, INTENT(IN) :: ntransient, nplot
DOUBLE PRECISION :: x,f
INTEGER :: ix

  x = x0
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!  Discard the first ntransient point to ensure fixed point is reached   !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  DO ix = 1, ntransient
    x = f(x,r)
  ENDDO

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!  Output the fixed points                                               !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  DO ix = 1, nplot
    x = f(x,r)
    CALL DOT (NINT((r-r0)*xscale)+25,425-NINT(x*yscale))
  ENDDO
END SUBROUTINE Output
```

```
DOUBLE PRECISION FUNCTION f (x,r)
DOUBLE PRECISION, INTENT(IN) :: x,r

  f = 4.d0*r*x*(1.d0-x)

END FUNCTION f
```

Input different function here

Chaos ($r = 1$) - x_n vs iteration n

Bufication



Universal Properties & Self-Similarity

- A sequence of period-doublings accumulating to a chaotic trajectory of infinite period at

  $$r = r_\infty = 0.892486417967\ldots$$

Bufication

- For $r > r_\infty$, the trajectory is very sensitive to the initial conditions. We also found "windows" of period 3, 6, 12, … embedded in the broad regions of chaotic behaviour.
- ***Such behavior is independent of the details of the form f(x).***
- Examples: $f(x) = x\, e^{r\,(1-x)}, \quad f(x) = r \sin \pi x.$

Quantitative Description

- Define b_k the value of r at which the fixed point of $f^{(2^{(k-1)})}(x)$ bifurcates and becomes unstable.
- e.g., $b_1 = 0.75$ and
  $b_2 = (1 + \sqrt{6})/4$.
- The distance between b_k becomes narrower and narrower, is there anyway to estimate it easily?

k	b_k
1	0.750 000
2	0.862 372
3	0.886 023
4	0.891 102
5	0.892 190
6	0.892 423
7	0.892 473
8	0.892 484

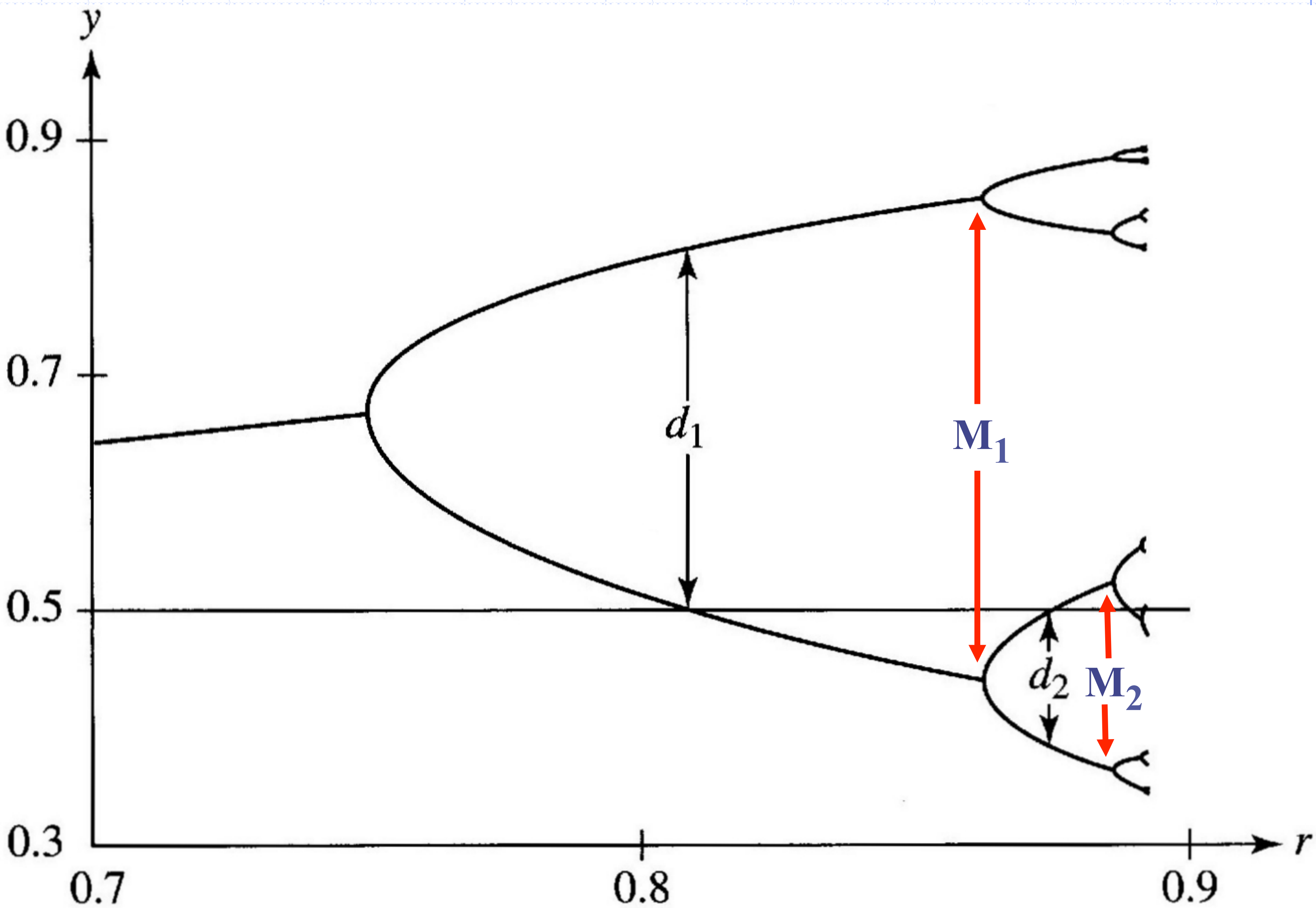
Critical Exponents

- More quantitatively, we define the Feigenbaum number, $\delta = \lim_{k\to\infty} \dfrac{b_k - b_{k-1}}{b_{k+1} - b_k}$.
- If δ exists, we have $b_k \approx r_\infty$ - $C\delta^{-k}$.
  For the logistic map,
  $\delta = 4.669201609102991\ldots$
- A better way of defining δ is to replace $\boldsymbol{b_k}$ by $\boldsymbol{s_k}$, where $\boldsymbol{s_k}$ are values of r that give superstable trajectories of period 2^{k-1}. (Project 6.1/6.22)

Critical Exponents

- We can also associate other quantities with the series of pitchfork bifurcations. One is the maximum distance $\boldsymbol{M_k}$ between the values of x describing the bifurcation. The other one is $\boldsymbol{d_k} = x^*$ - ½, where x^* is the value of the fixed point nearest to the fixed point $x^* = 1/2$ and

$$\alpha = -\lim_{k\to\infty}\left(\frac{d_k}{d_{k+1}}\right) = 2.502907875095892 8485...$$


y
0.9
0.7
0.5
0.3
0.7
0.8
0.9
r
d_1
M_1
d_2
M_2

Critical Exponents

- A better way of defining δ is to replace $\boldsymbol{b}_k$ by $\boldsymbol{s}_k$, where $\boldsymbol{s}_k$ are values of r that give superstable trajectories of period 2^{k-1}.
- One find $\boldsymbol{s}_k = \boldsymbol{r}$ by solving the equation

$$f^{(k)}(x=½) = 1/2,$$

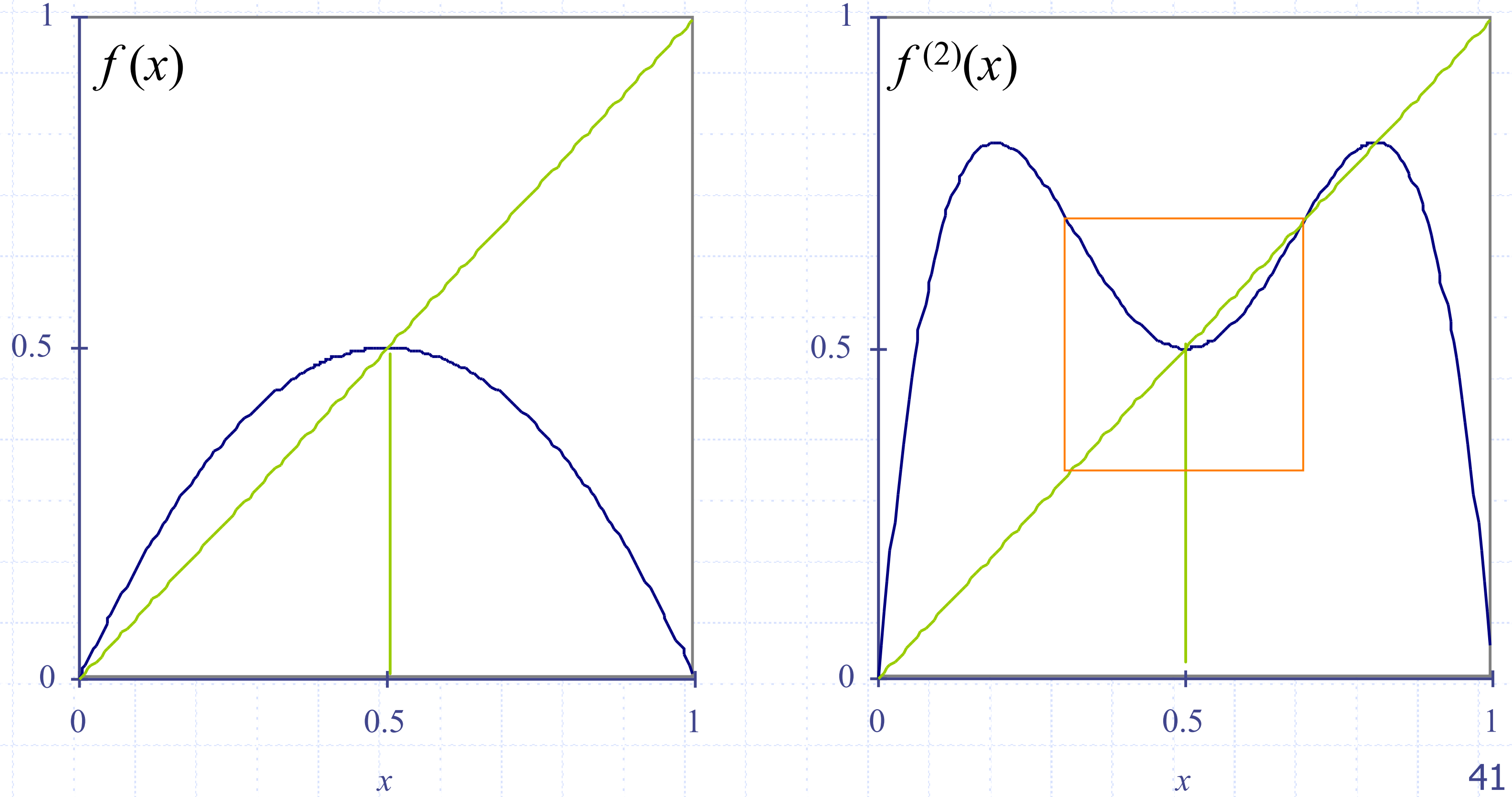
with period 2^{k-1}

k	Period	s_k
1	1	0.500 000 000
2	2	0.809 016 994
3	4	0.874 640 425
4	8	0.888 660 970
5	16	0.891 666 899
6	32	0.892 310 883
7	64	0.892 448 823
8	128	0.892 478 091

Examples: $8r^2(1-r)=1$ leads to $s_1=1/2$, $s_2=(1+\sqrt{5})/4$, etc.

Self-Similarity

Period-Doubling is characterised by self-similarities, e.g., the period-doublings look similar except for a change of scale (***renormalization***).



Self-Similarity

Such behavior suggests that the quantities α and δ are ***universal***, i.e., independent of the detailed form of $f(x)$. (r_∞ and C do.) This is true for maps that have a quadratic maximum (shown by Feigenbaum).

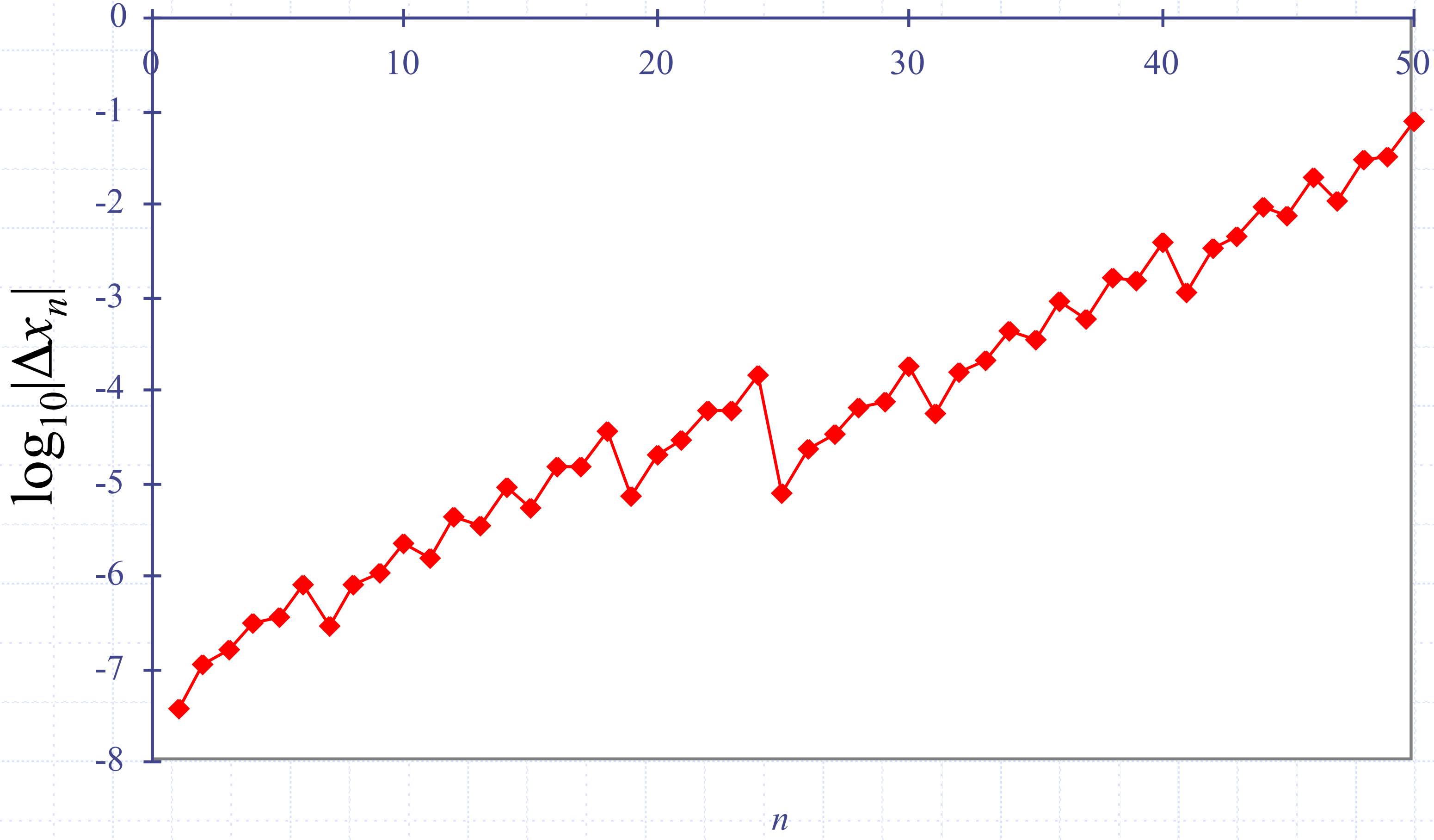
$$\boldsymbol{\delta = 4.669\ 201\ 609\ 102\ 991\ \ldots}$$

$$\boldsymbol{\alpha = 2.502\ 907\ 875\ 095\ 892\ 848\ 5\ldots}$$

Measuring Chaos

- The most important characteristic of chaos is ***sensitivity to initial conditions***, which limits our ability to make numerical predictions of the trajectories even though the logistic map is deterministic.
- Quantitatively, we measure Δ_n, the difference between trajectories of two identical dynamical systems from different initial conditions as function of n.

The evolution of the difference Δx_n between trajectories of the logistic map at $r = 0.91$ for $x_0 = 0.5$ and $x_0 = 0.5001$.



Lyapunov Exponent (Research Project)

- Assuming $|\Delta_n| = |\Delta_0|\, e^{\lambda n}$
  where λ is called the ***Lyapunov*** exponent.
- If λ is positive then nearby trajectories diverge ***exponentially*** and the system is chaotic.
- One can obtain λ naively or do better by reformatting

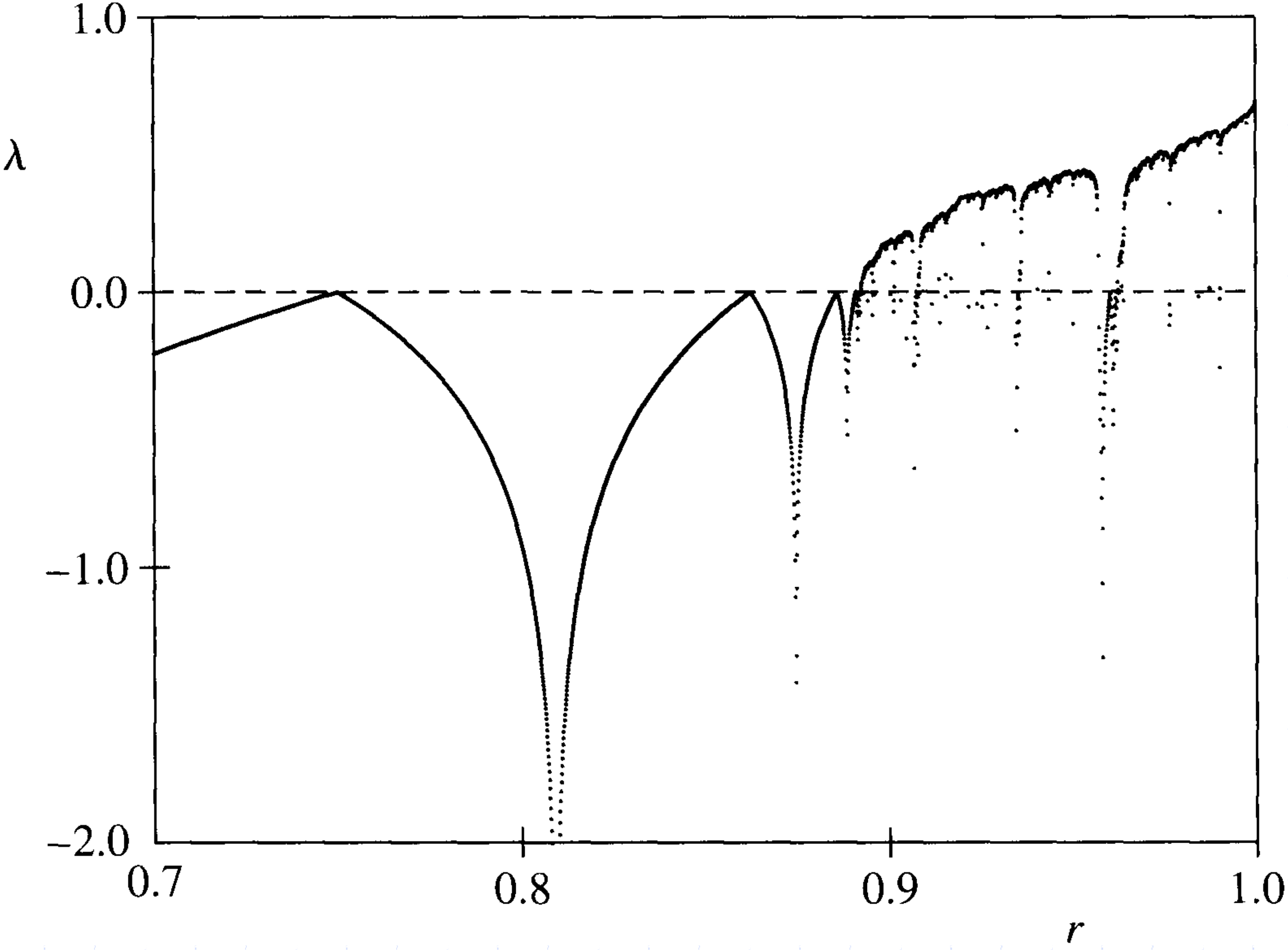
$$\lambda = \frac{1}{n}\ln\left|\frac{\Delta x_n}{\Delta x_0}\right|, \quad \text{write } \frac{\Delta x_n}{\Delta x_0} = \frac{\Delta x_1}{\Delta x_0}\frac{\Delta x_2}{\Delta x_1}\cdots\frac{\Delta x_n}{\Delta x_{n-1}}$$

$$\lambda = \frac{1}{n}\sum_{i=0}^{n-1}\ln\left|\frac{\Delta x_{i+1}}{\Delta x_i}\right|$$

- We can also replace $\Delta x_{i+1}/\Delta x_i$ by $dx_{i+1}/dx_i = f'(x_i)$

$$\lambda = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right|$$

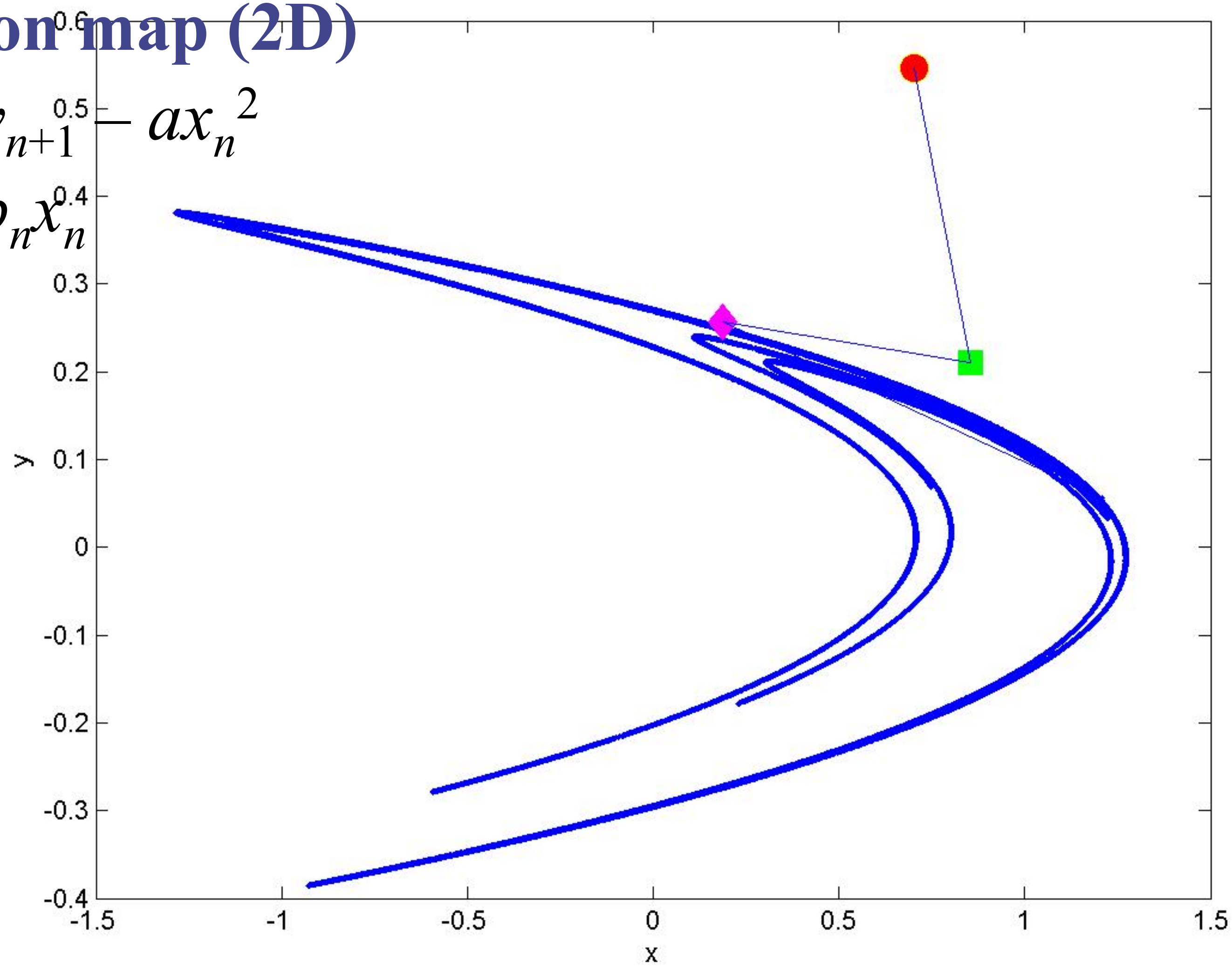
Lyapunov Exponent



Higher-Dimensional Models

The Hēnon map (2D)

$$x_{n+1} = y_{n+1} - a x_n^2$$
$$y_{n+1} = b_n x_n$$



Higher-Dimensional Models

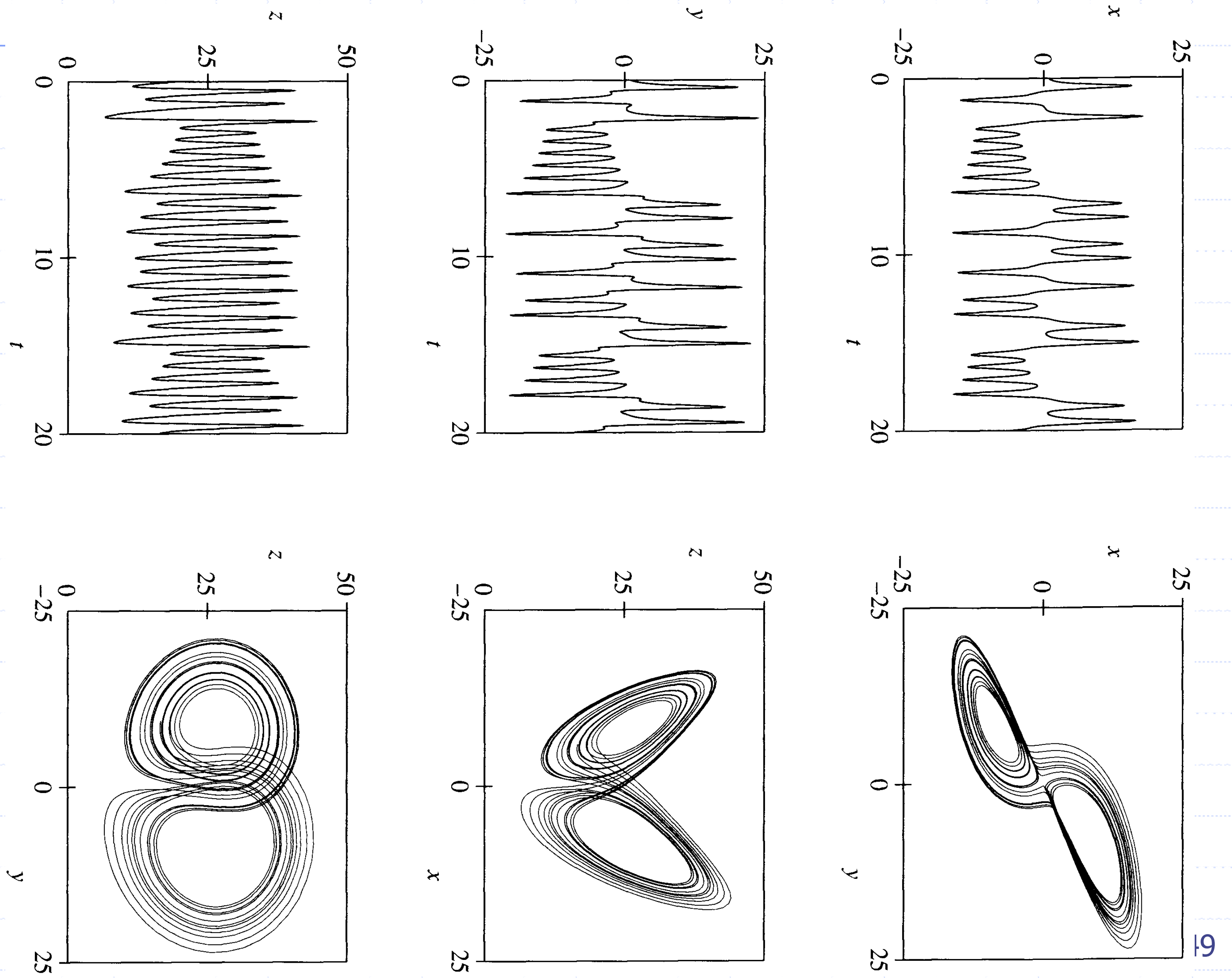
- **The Lorentz model (3D)**

$$\frac{dx}{dt} = -\sigma x + \sigma y, \quad \frac{dy}{dt} = -xz + rx - y, \quad \frac{dz}{dt} = xy - bz,$$

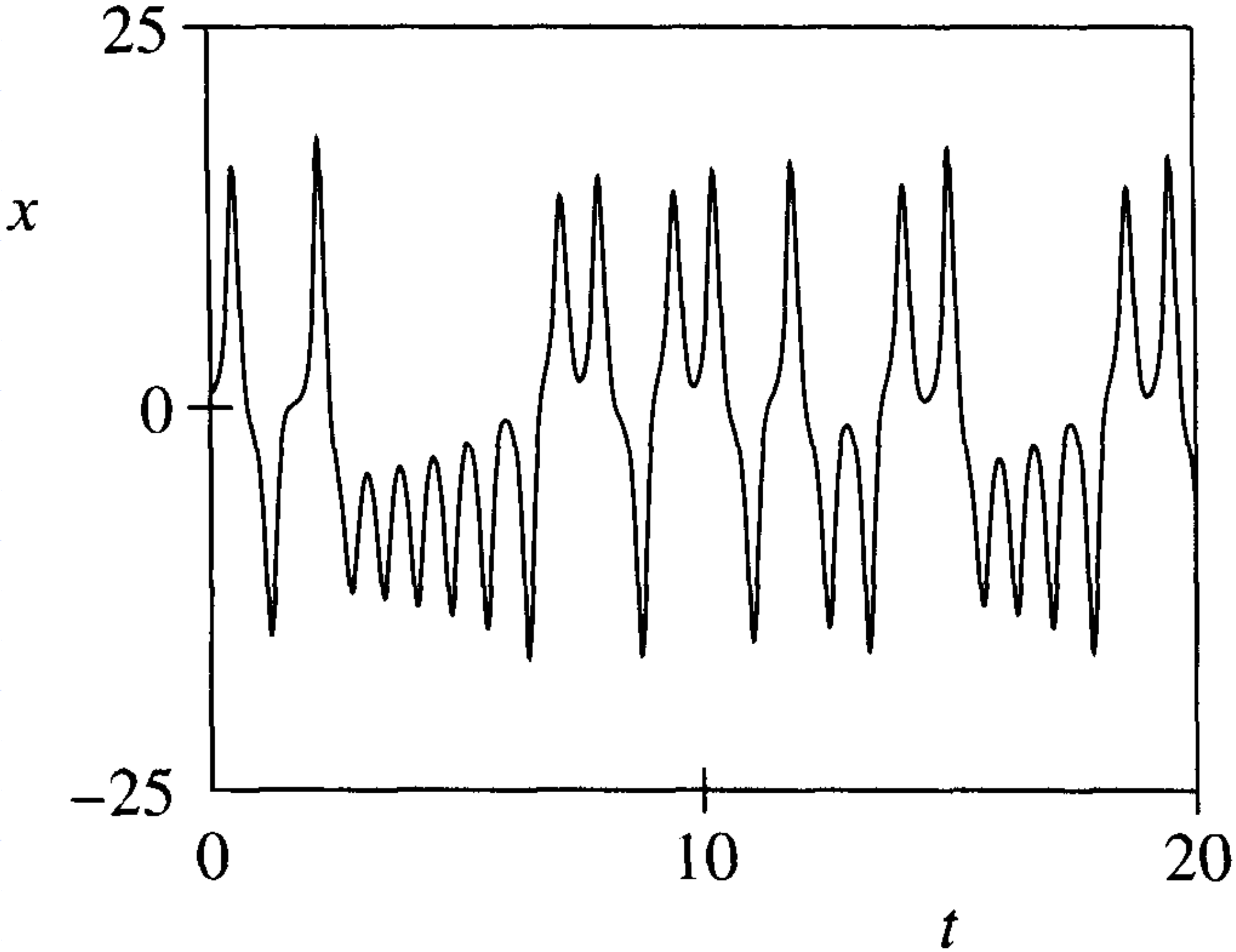
where x, y, z, σ, r and b are parameters for the Raleigh-Benard cell.

- The application of Lorentz equations to weather prediction has led to a popular metaphor known as the ***butterfly effect***.

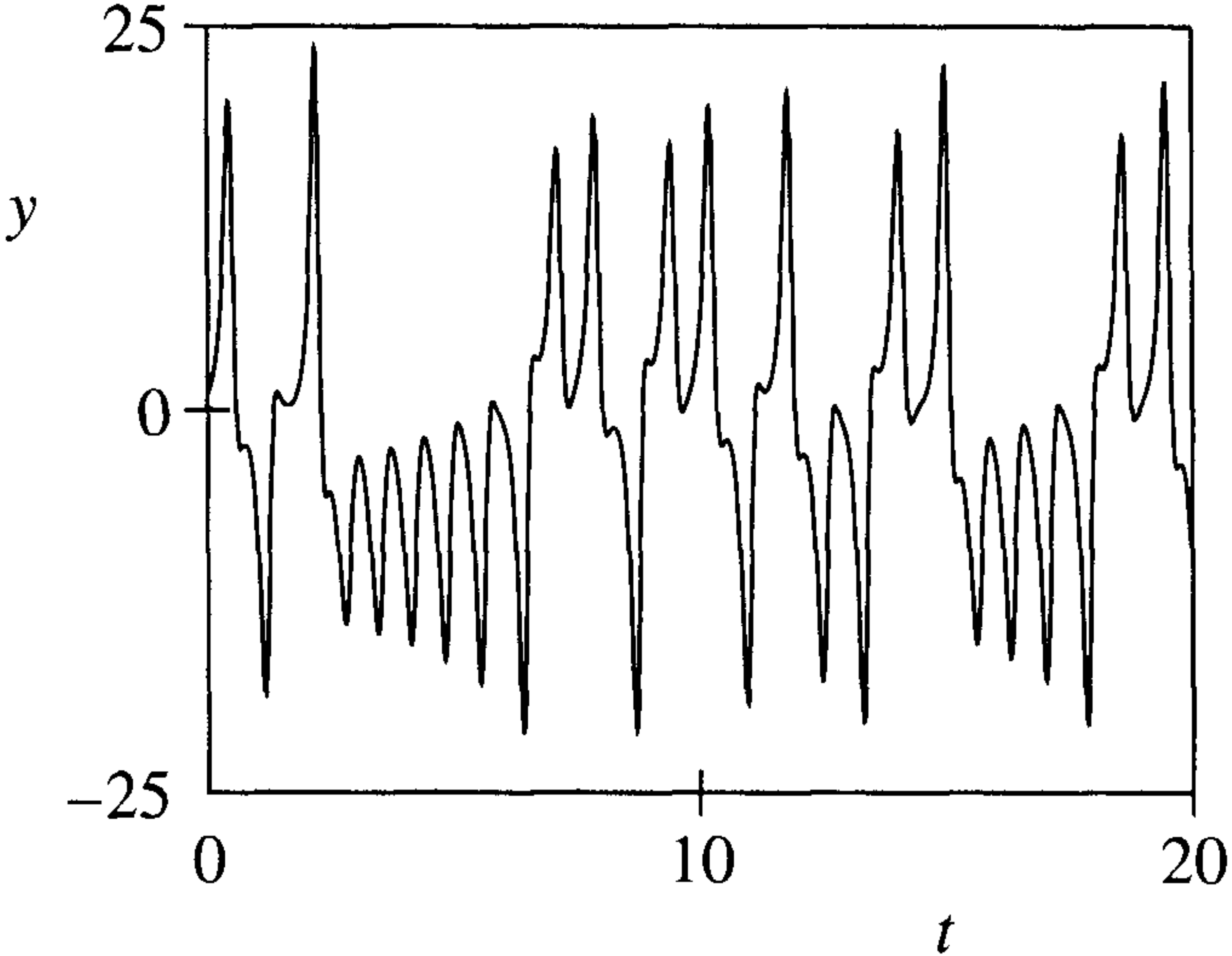
Lorenz Attractor



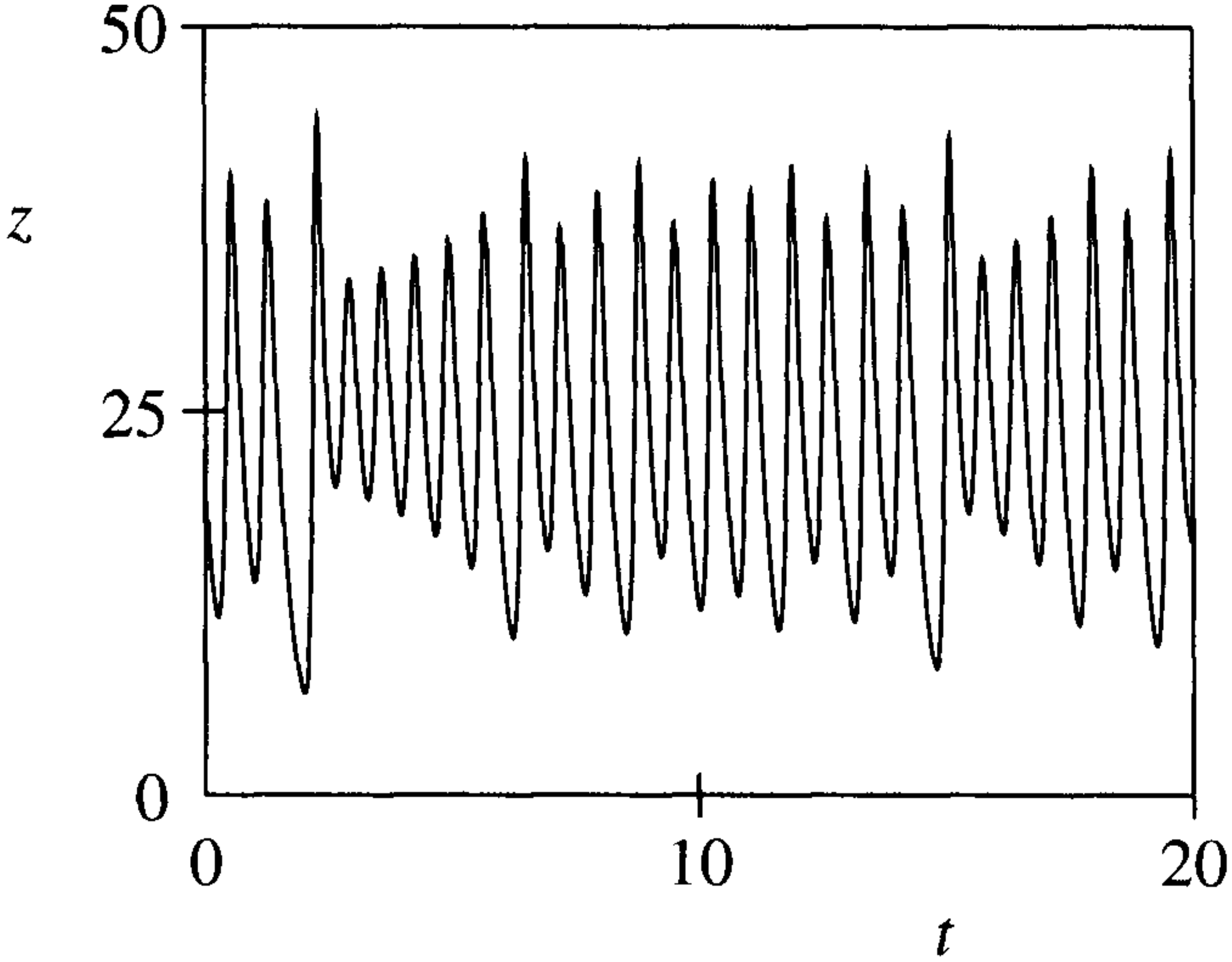
Lorenz Attractor



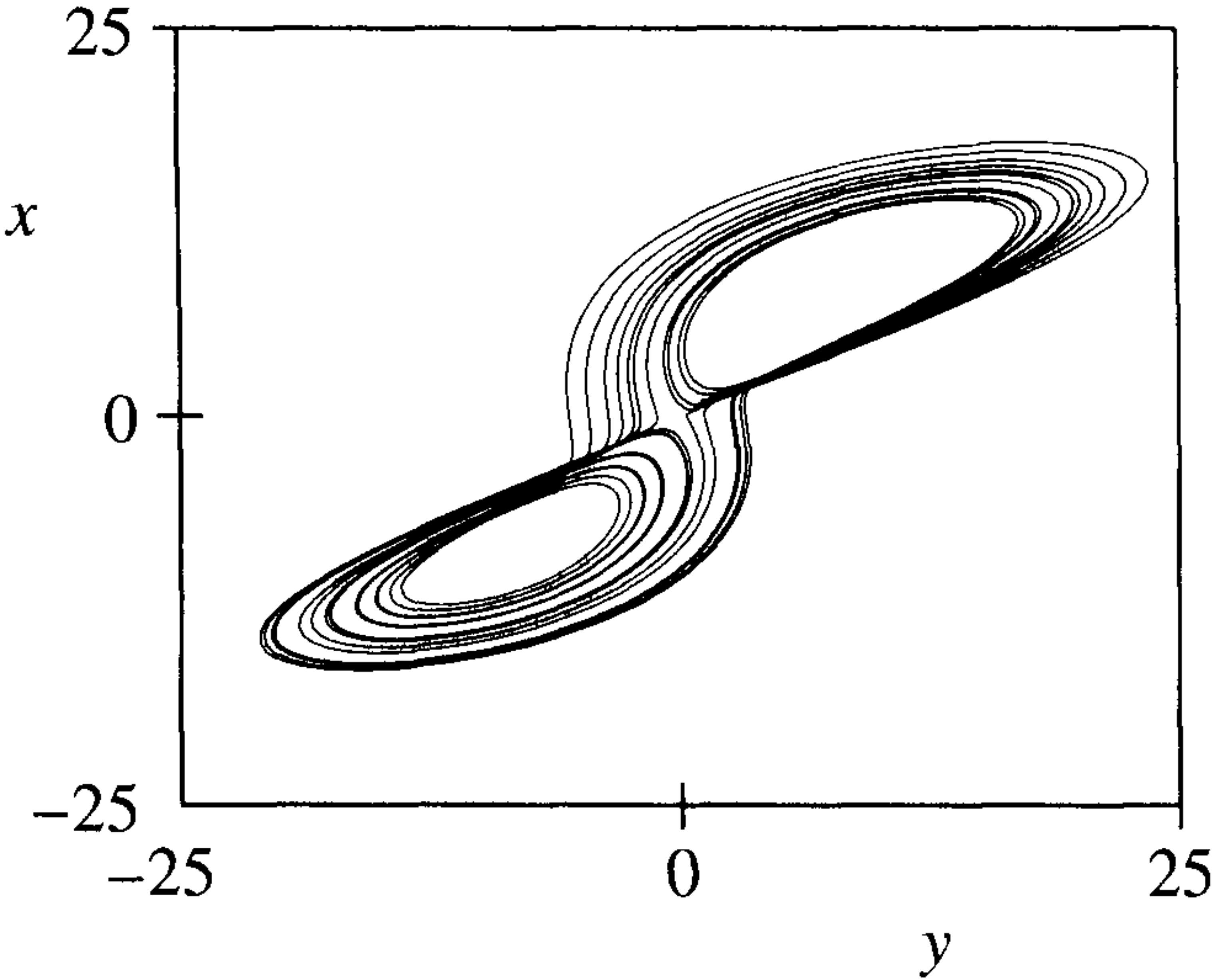
Lorenz Attractor



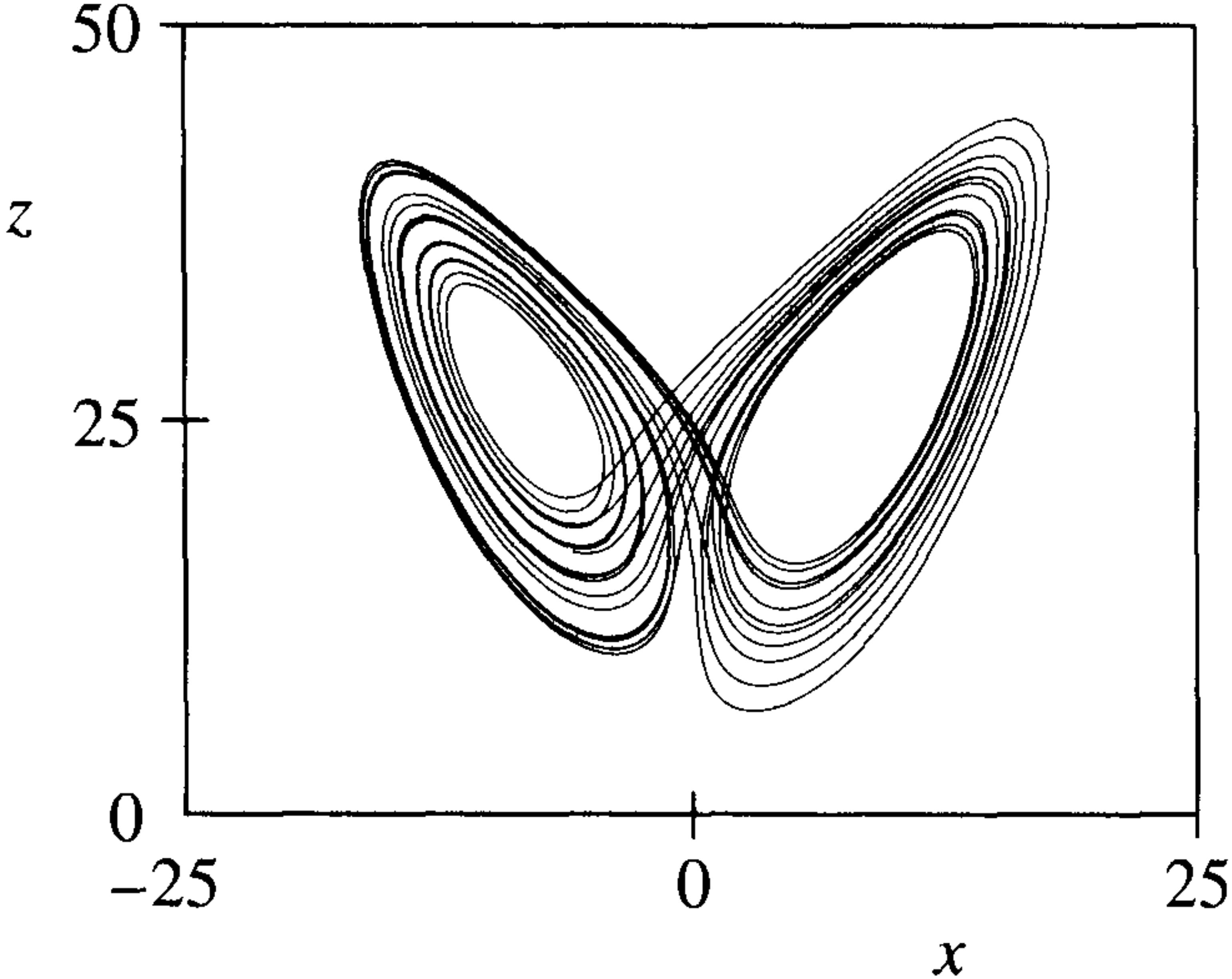
Lorenz Attractor



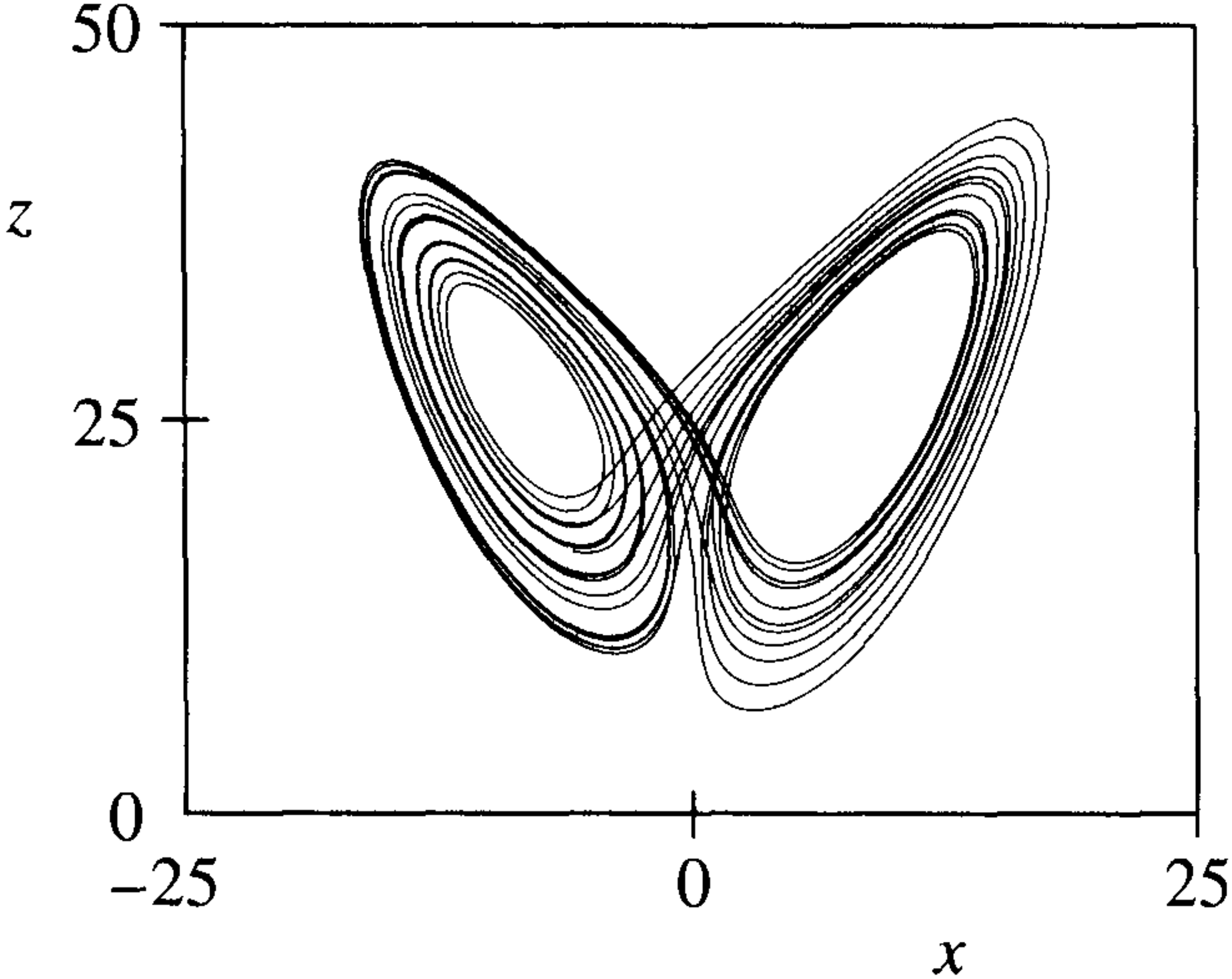
Lorenz Attractor



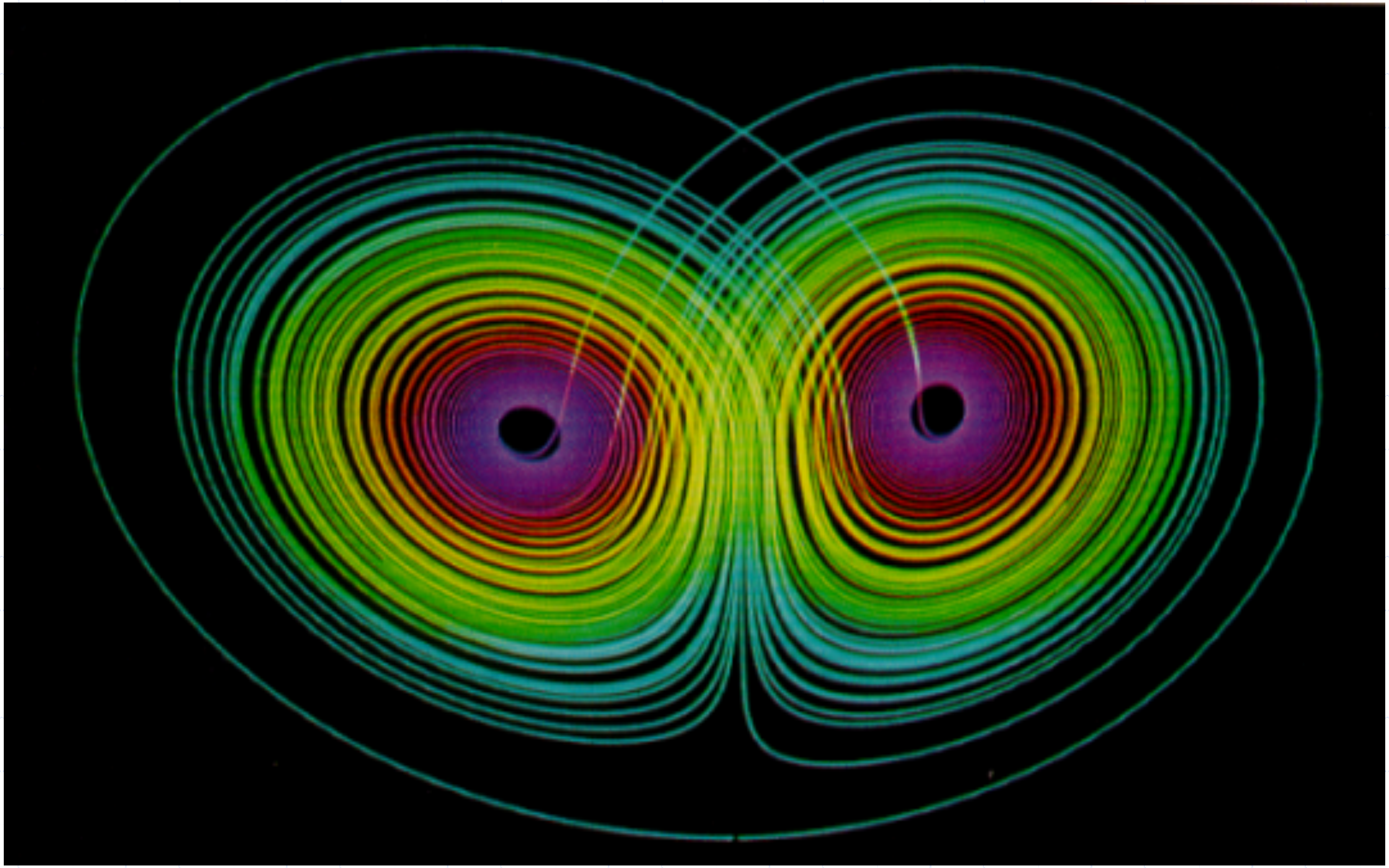
Lorenz Attractor



Lorenz Attractor



Lorenz Attractor



Forced Damped Pendulum

- We consider the dynamics of a non-linear mechanical system, the simple pendulum, described by

$$\frac{d^2\theta}{dt^2} = -\gamma \frac{d\theta}{dt} - \left[\omega_0^2 + 2A\cos\omega t\right]\sin\theta,$$

where θ is the angle the pendulum makes with the vertical axis, γ is the damping coefficient, $\omega_0{}^2 = g/L$ is the natural frequency of the pendulum, and ω and A are the frequency and amplitude of the external force.

Forced Damped Pendulum

- There exist a stable fixed point $x = 0$, $v = 0$ for sufficiently small A.
- As A increase, this attractor becomes unstable.
- We study a so called ***Poincare map*** by plotting a point in phase space after every cycle of the external force, i.e. we plot $d\theta/dt$ versus θ for values of $t = nT$, $T = 2\pi/\omega$.
- Program **poincare** uses 4th-order Runge-Kutta algorithm to Compute θ and the angular velocity $d\theta/dt$ for the pendulum. See p.157 for program details.

Gap Bifurcations in Nonlinear Dynamical Systems

F. B. Rizzato[1,*] and R. Pakter[1,2,3]

[1]*Instituto de Física, Universidade Federal do Rio Grande do Sul, P.O.Box 15051, 91501-970, Porto Alegre, RS, Brasil*
[2]*Departamento de Matemática, Universidade Luterana do Brasil, Av. Miguel Tostes 101, 92420, Canoas, RS, Brasil*
[3]*Universidade Estadual do Rio Grande do Sul, Rua Inconfidentes 395, 93340-000, Novo Hamburgo, RS, Brasil*
(Received 29 March 2002; published 10 October 2002)

We investigate the dynamics generated by a type of equation which is common to a variety of physical systems where the undesirable effects of a number of self-consistent nonlinear forces are balanced by an externally imposed controlling harmonic force. We show that the equation presents a new sequence of bifurcations where periodic orbits are created and destroyed in such a nonsimultaneous way that may leave the appropriate phase-space occasionally empty of fundamental harmonic orbits and confined trajectories. A generic analytical model is developed and compared with a concrete physical example.

$$\frac{d^2r}{ds^2} + k^2(s)\,r = F(r). \qquad (1)$$

$k(s)$ is a periodic function satisfying $k(s+1) = k(s)$, with time average $\overline{k^2(s)} = k_0^2$; k_0 is constant and the period has been normalized to the unity. $F(r)$ contains all nonlinear forces acting on the system.

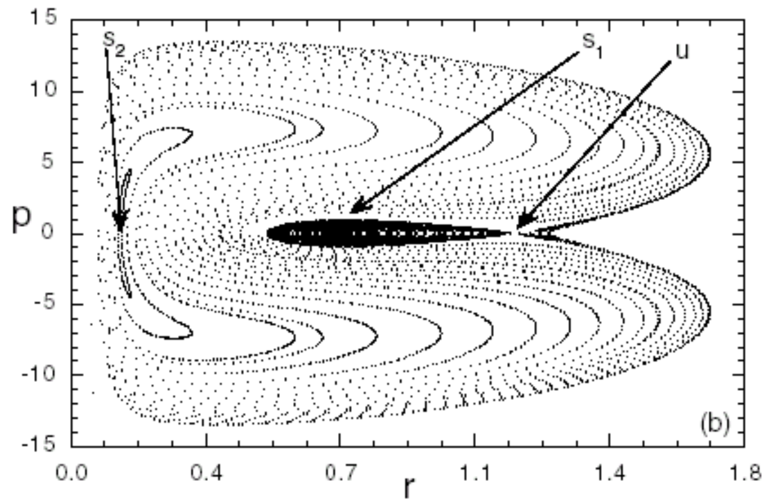


FIG. 2. A comparison of ν curves with the phase plot using the same parameters as Fig. 1(a).

Double Pendulum

$$x_1 = L\sin\theta_1$$

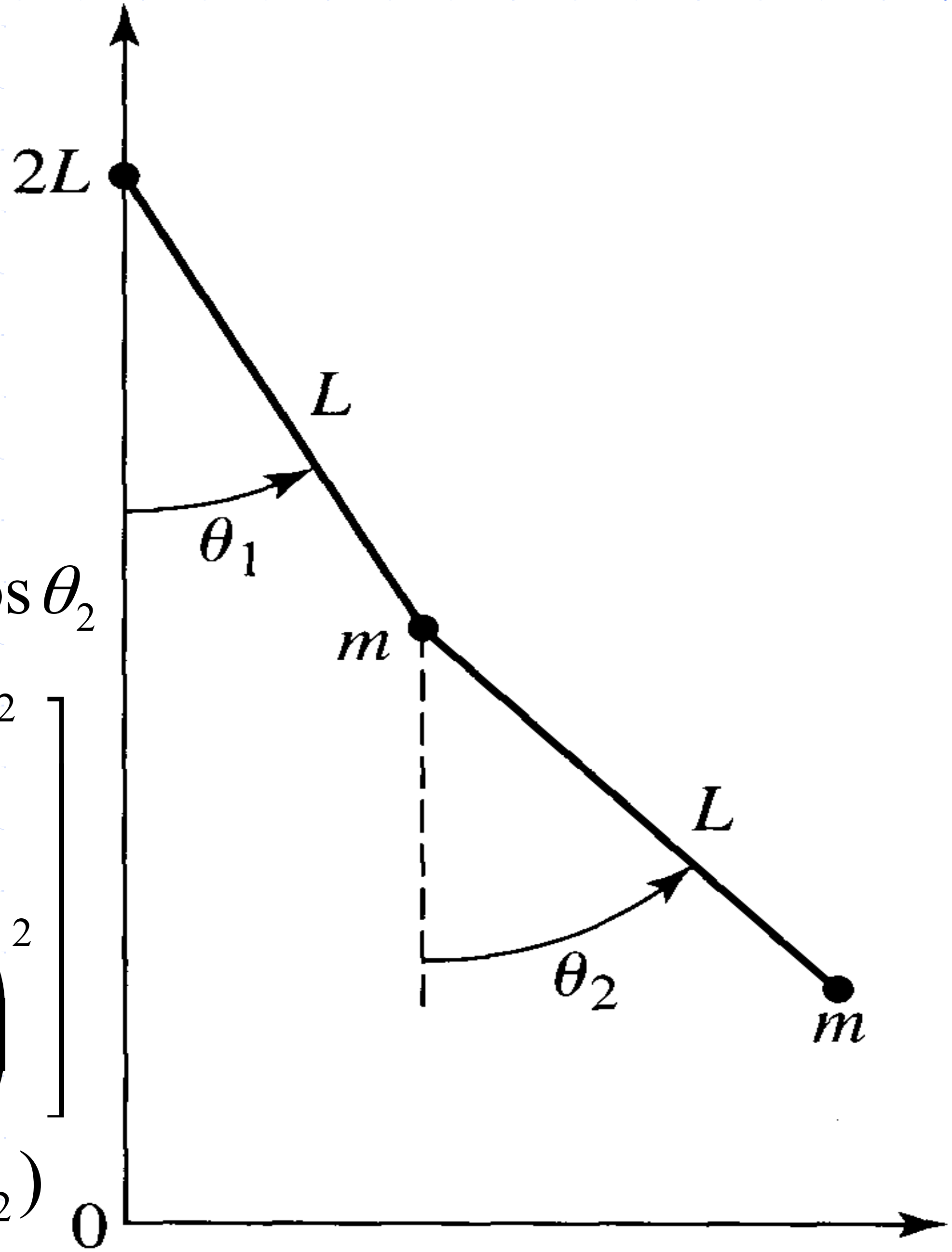
$$y_1 = 2L - L\cos\theta_1$$

$$x_2 = L\sin\theta_1 + L\sin\theta_2$$

$$y_2 = 2L - L\cos\theta_1 - L\cos\theta_2$$

$$K = \frac{1}{2}m\left[\left(\frac{dx_1}{dt}\right)^2 + \left(\frac{dy_1}{dt}\right)^2 + \left(\frac{dx_2}{dt}\right)^2 + \left(\frac{dy_2}{dt}\right)^2\right]$$

$$U = mgL(3 - 2\cos q_1 - \cos q_2)$$



Double Pendulum

Generalized coordinates and momenta:

$$q_1 = \theta_1, \quad q_2 = \theta_2$$

$$p_1 = \partial K / \partial \dot{q}_1, \quad p_2 = \partial K / \partial \dot{q}_2$$

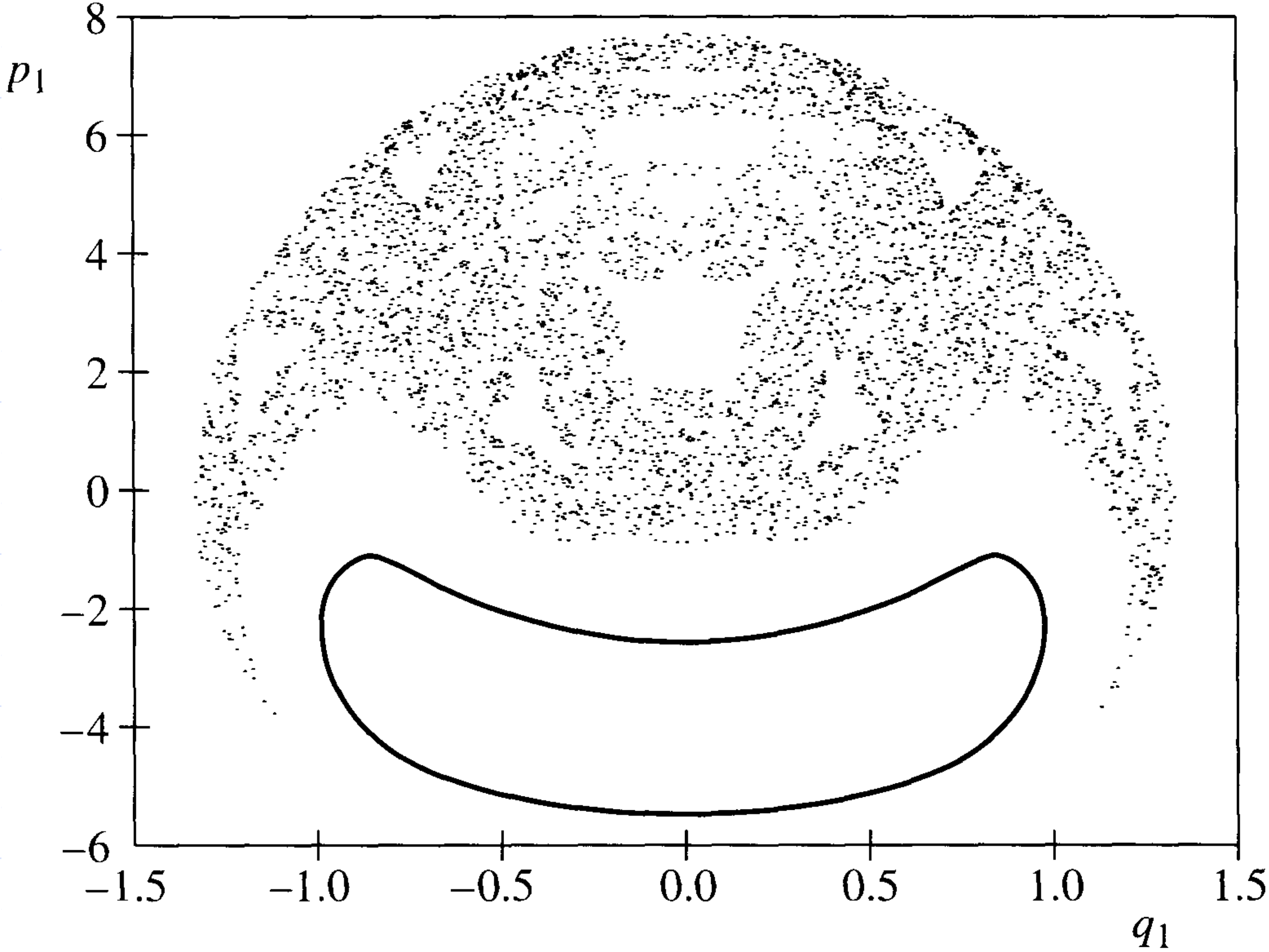
$$H = \frac{1}{2mL^2} \frac{p_1^2 + 2p_2^2 - 2p_1 p_2 \cos(q_1 - q_2)}{1 + \sin^2(q_1 - q_2)}$$

$$+ mgL(3 - 2\cos q_1 - \cos q_2)$$

Equation of motion (used as iteration scheme):

$$\dot{p}_i = \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \quad \dot{q}_i = \frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$

Double Pendulum



Lecture 8 Review & Required

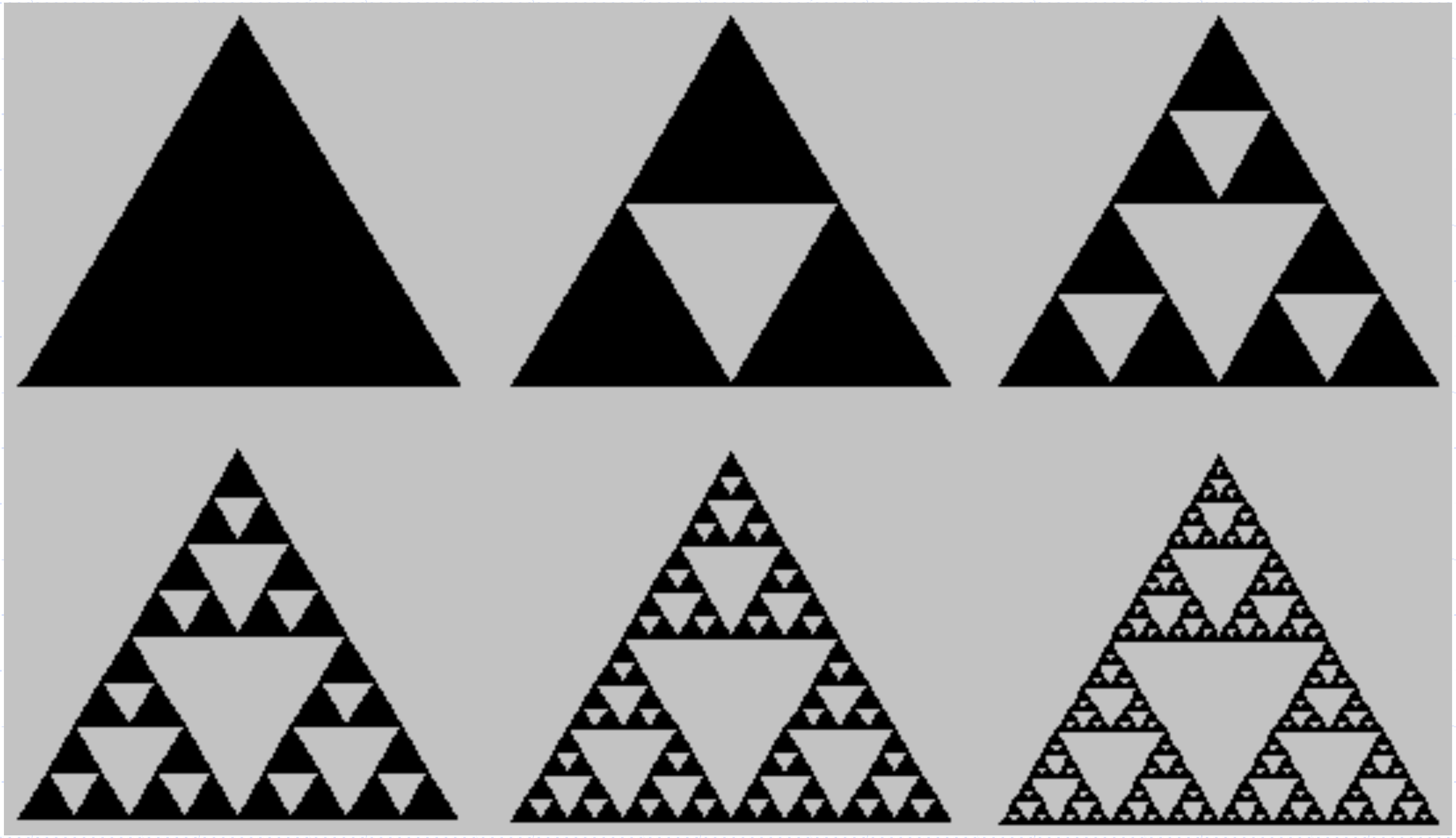
Simple systems can exhibit complex behaviour

- **The logistic map.**
- **Fixed point (attractor), stable/unstable/superstable.**
- **Bifurcation, period-doubling, chaos.**
- **Self-similarity (renormalization), universality, critical exponents.**
- Program: **logistic, bifurcate**, graph_sol, poincare, period.
- Graphics method, recursion, (bisection).
- Controlling chaos (optional).
- Quantum chaos, **Projects**, etc.

Fractal

- **A classic book to read:**
  The Fractal Geometry of Nature
  by Benoit Mandelbrot
  (Freeman, New York, San Francisco, 1982)
- ``Clouds are not spheres, mountains are not cones, coastlines are not circles, bark is not smooth, nor does lightning travel in a straight line.''
- Introduced the term **``Fractal Geometry''.**
- **Random and Non-random Fractals**

Sierpinski triangle

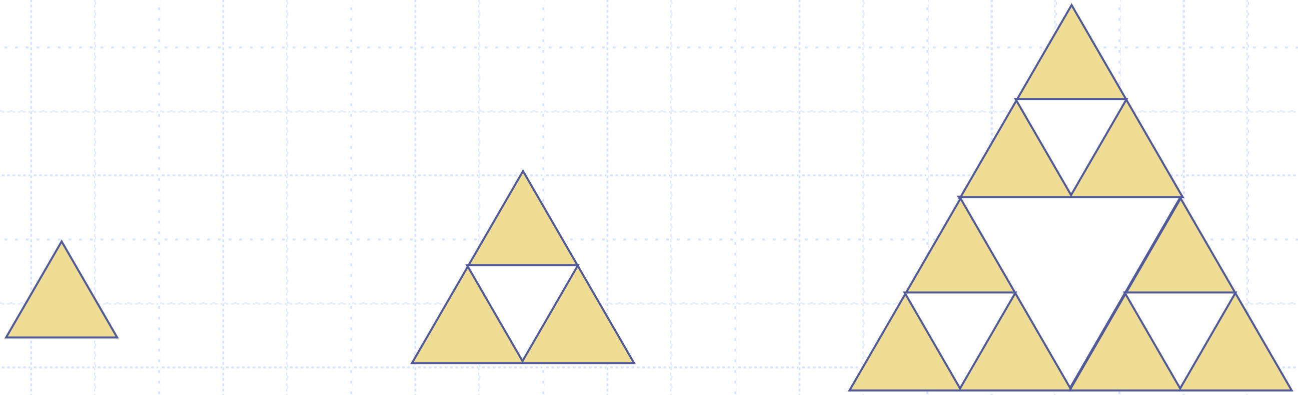


The Sierpinsky Gasket

- **Define a *Growth Rule* with a *basic unit*, triangular shaped tile, `mass' *M*, `length' *L*.**
- **Iterate** the growth rule over and over until you run out of tiles.
- Define the density $\rho(L)$ as the fraction of space covered by black tiles, as a function of L:

$$\rho(L) = M(L)/L^2$$

The Sierpinsky Gasket



(a) (b) (c)

$L = 2^0$ $\quad$ $L = 2^1$ $\quad$ $L = 2^2$

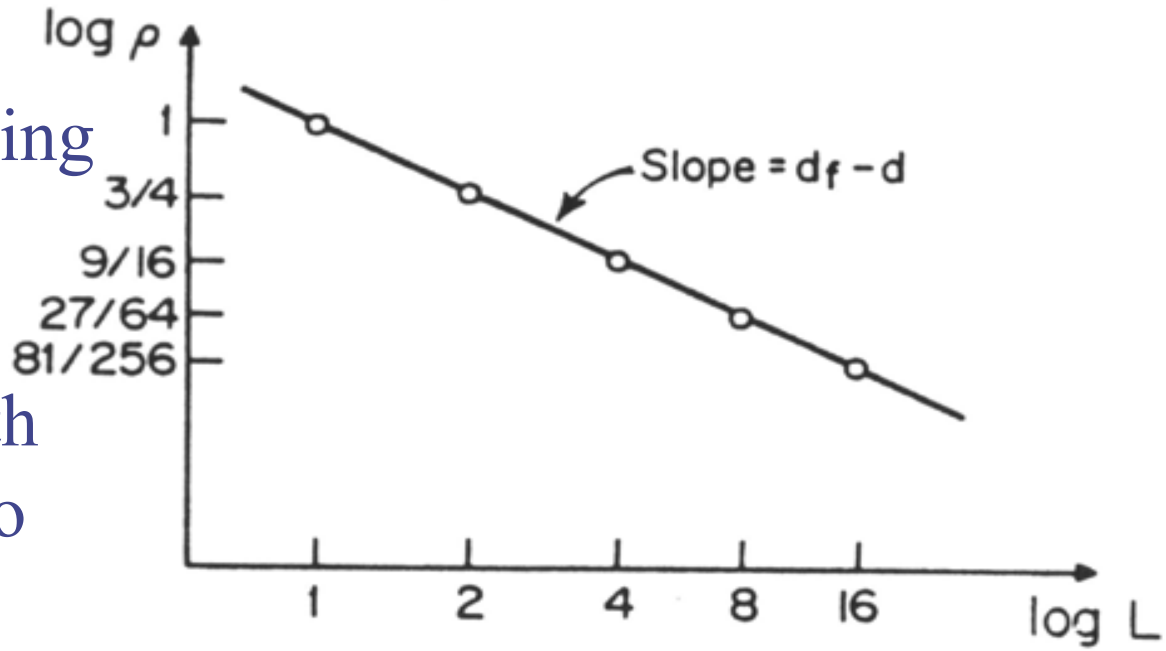
$M = 3^0$ $\quad$ $M = 3^1$ $\quad$ $M = 3^2$

$$\rho = \left(\frac{3}{4}\right)^0 \qquad \rho = \left(\frac{3}{4}\right)^1 \qquad \rho = \left(\frac{3}{4}\right)^2$$

The Sierpinsky Gasket

Displays two striking features:

$\rho(L)$ decreases monotonically with L, without limit, so that by iterating sufficiently we can achieve an object of as low a density as we wish



$\rho(L)$ decreases with L in a predictable fashion, namely a simply power law, $\rho(L) = L^2$

The Fractal Dimension d_f

$$\alpha = slope = \frac{\log 1 - \log(3/4)}{\log 1 - \log 2} = \frac{\log 3}{\log 2} - 2$$

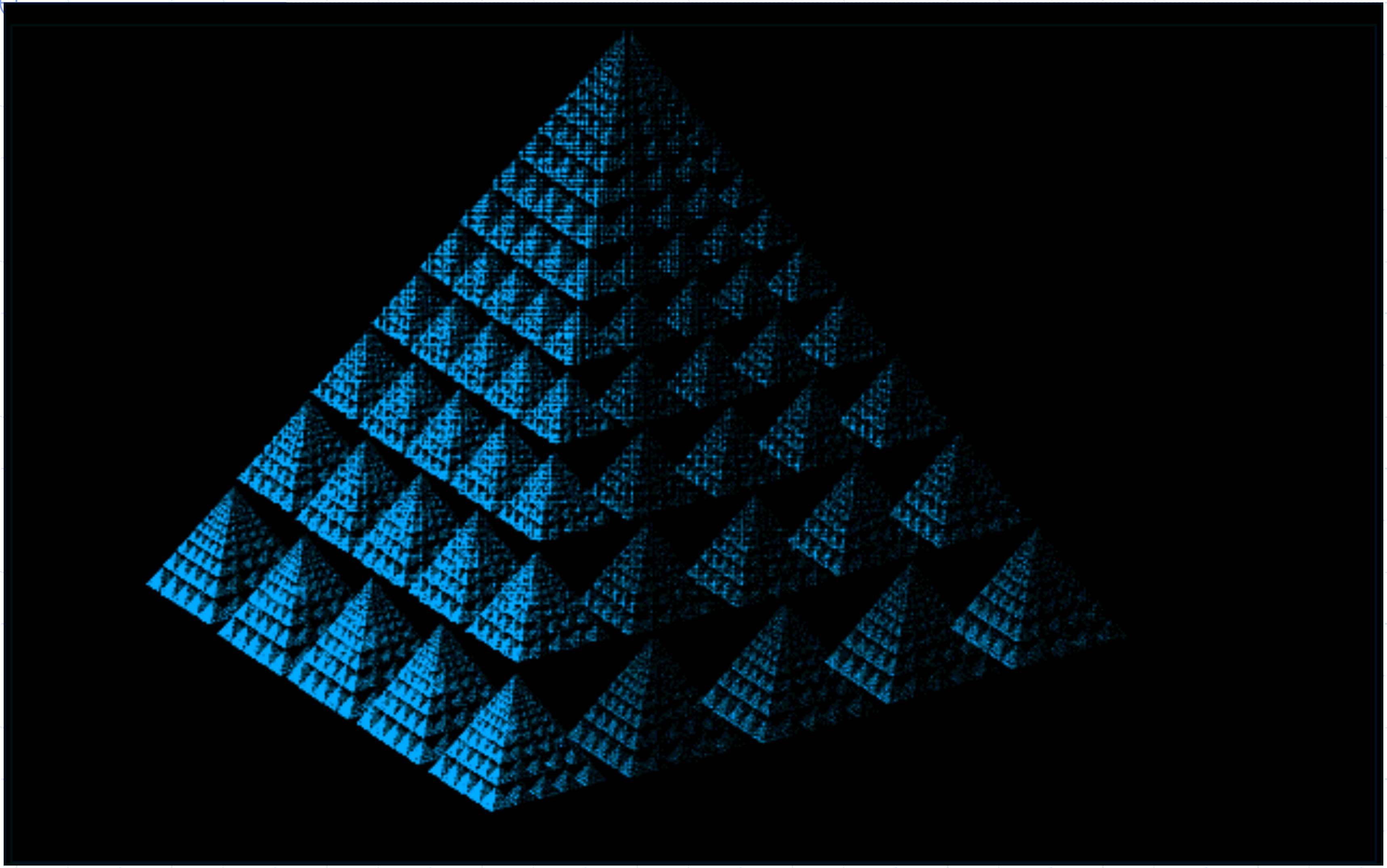
$$M(L) \equiv AL^{d_f}$$

$$\rho(L) = M(L)/L^2 = AL^{d_f - 2}$$

$$d_f = \log 3 / \log 2 = 1.58\ldots$$

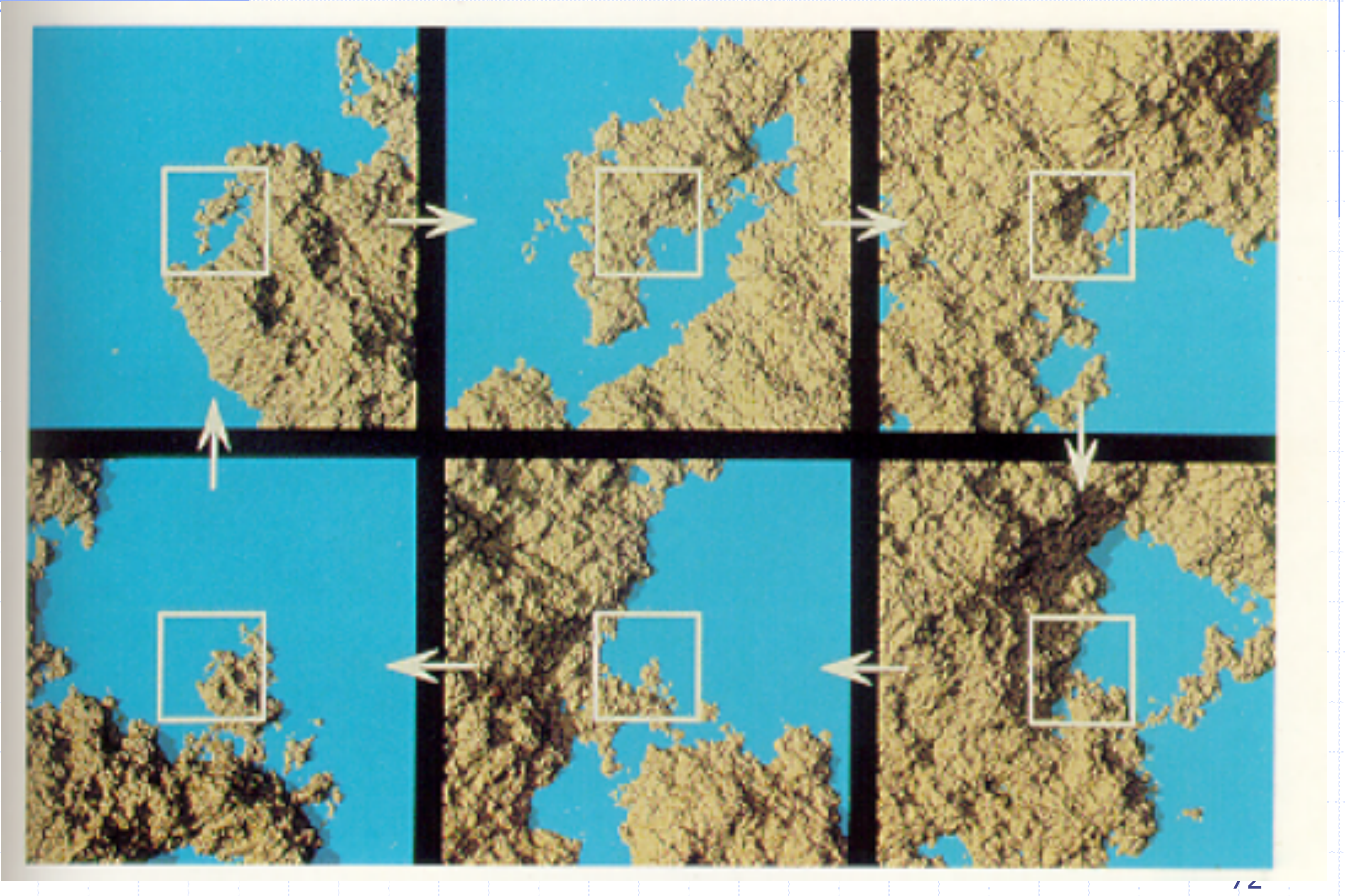
The Sierpincki gasket has a dimension intermediate between that of a line and a square, *fraction dimension*.

Sierpinski pyramid

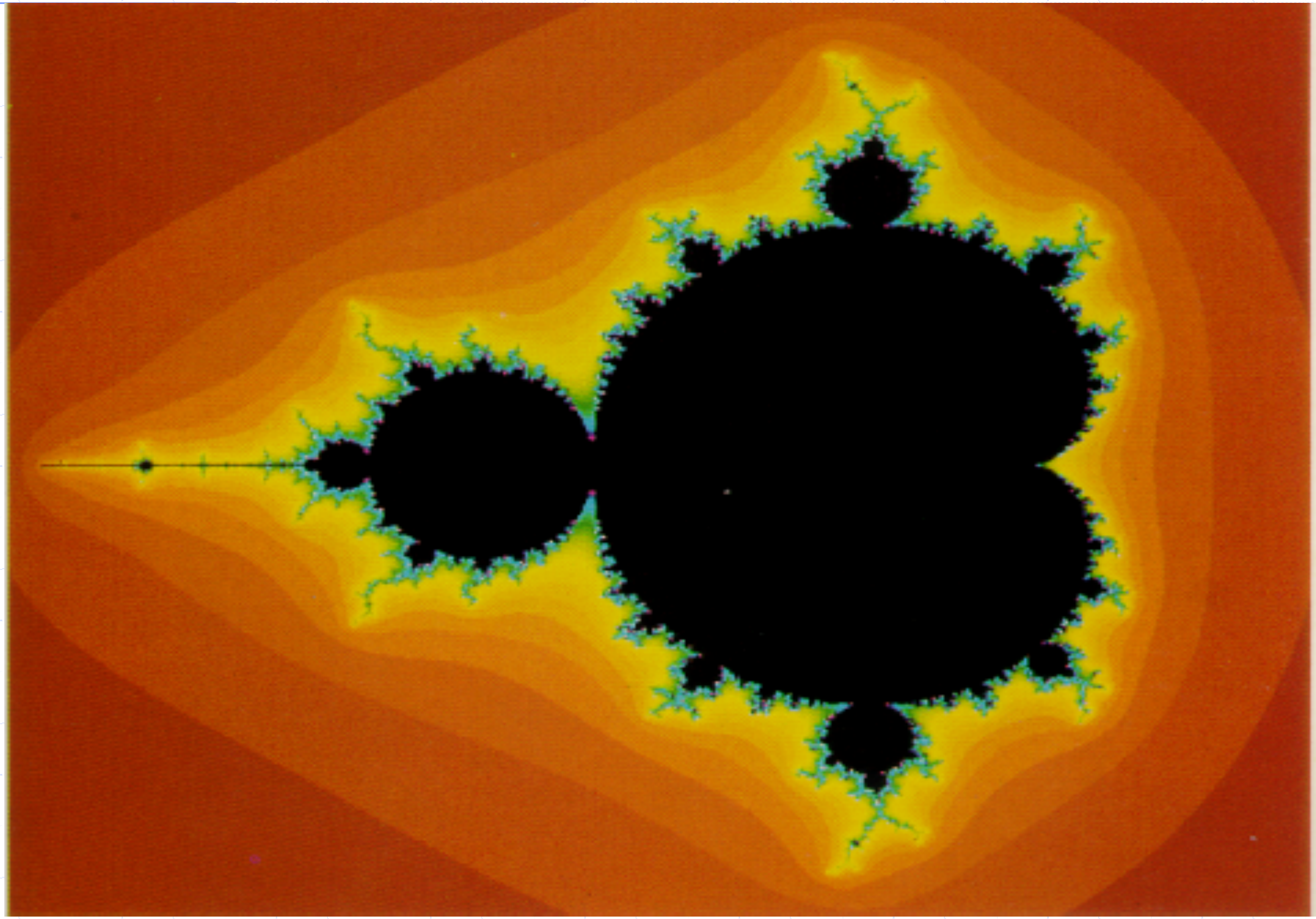


Fractals in nature

Coastline



Mandelbrot set

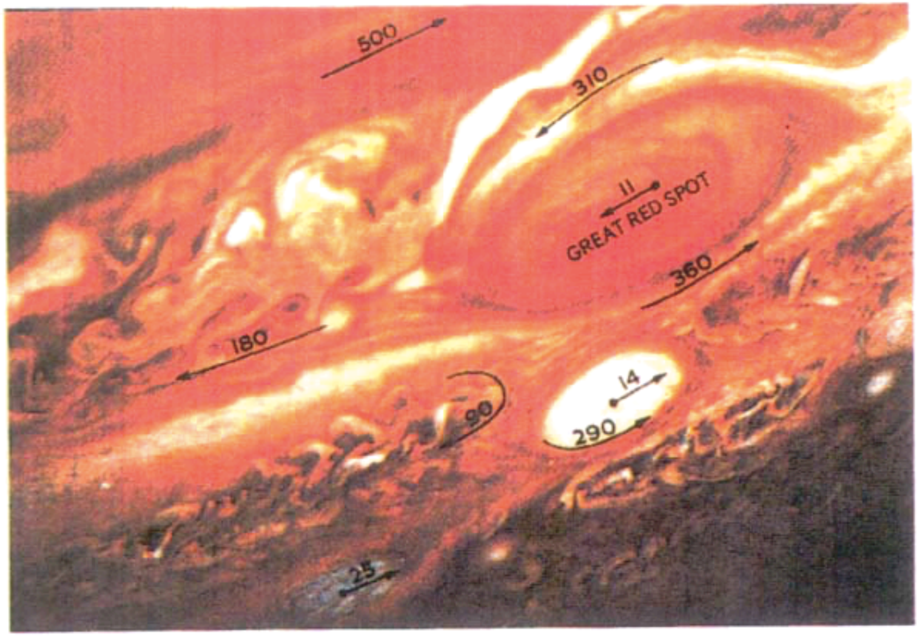


Other Examples

- Solar System
- Weather forecasting
- Stock Market

Jupiter

The atmosphere of Jupiter, as seen in the famous N.A.S.A. Voyager images. Prominent are the Great Red Spot and numerous other coherent long-lived features of the Jovian atmosphere. Note also the highly disorganized and time-varying flow coexisting with the large-scale order. (Plates courtesy of N.A.S.A.)



Characters

- Generated by Simple Rules
- Deterministic
- Irregular and Unpredictable
- Self-similarity
- Fractional Dimension
- Infinity of Scales

Controlling Chaos (Optional)

- *Can we control chaotic system's behaviour with small perturbation?*
- Note that for any value of r in the chaotic regime, there is an infinite number of trajectories that have *unstable periods*. Can a perturbation keeps the system on the desired unstable periodic trajectory?

1st Step: Find the unstable periodic trajectory $x(i)$, $i = 1$ to p, for the desired value of r_0. This is equivalent to finding a fixed point of the map $f^{(p)}$, i.e., solving the equation

$$g^{(p)}(x^*) \equiv f^{(p)}(x^*) - x^* = 0$$

- We need to apply ***root finding algorithms*** to this equation. The simplest one is the ***bisection*** method.

Controlling Chaos (Optional)

2nd Step: Iterate the map with $r = r_0$ until x_n is within ε of $x(1)$. Then use equation below to determined r.

- Suppose that we want to stabilize the unstable trajectory of period p for a choice of $r = r_0$. The idea is to make small adjustments of $r = r_0 + \Delta r$ at each iteration so that the difference between actual trajectory and the target periodic trajectory is small.
- Assume the actual trajectory is x_n and we with the trajectory to be at $x(i)$, then, to first-order:

Controlling Chaos (Optional)

$$x_{n+1} - x(i+1) = f(x_n, r) - f(x(i), r_0)$$

$$= \frac{\partial f(x,r)}{\partial x}\left[x_n - x(i)\right] + \frac{\partial f(x,r)}{\partial x}\Delta r = 0$$

$$4r_0\left[1-2x(i)\right]\left[x_n - x(i)\right] + 4x(i)\left[1-x(i)\right]\Delta r = 0.$$

$$\Delta r = -r_0\frac{\left[1-2x(i)\right]\left[x_n - x(i)\right]}{x(i)\left[1-x(i)\right]}.$$

- Iterate the logistic map at $r = r_0$ until x_n is sufficiently close to $x(i)$. Then use the above equation to change the value of r so that the next iteration is closer to $x(i+1)$.

3rd Step: To turn off the control, set $r = r_0$.

The Bisection Algorithm

Appendix 6B, root-finding

1. Choose two values, x_{left} and x_{right}, with $x_{left} < x_{right}$, such that $g^{(p)}(x_{left})\, g^{(p)}(x_{right}) < 0$. There must be a value of x such that $g^{(p)}(x) = 0$ in the interval $[x_{left}, x_{right}]$.
2. Chose the midpoint, $x_{mid} = x_{left} + ½\,(x_{right} - x_{left}) = ½\,(x_{right} + x_{left})$, as the guess for x^*.
3. If $g^{(p)}(x_{mid})$ has the same sing as $g^{(p)}(x_{left})$, then replace x_{left} by x_{mid}; otherwise, replace x_{right} by x_{mid}. Thus we halved the interval for the location of the root.
4. Repeat steps 2 and 3 until the desired level of precision is achieved.

```
DEF f(x,r,p)                                ! F defined by recursive procedure
  IF p > 1 then
    LET y = f(x,r,p-1)
    LET f = 4*r*y*(1-y)
  ELSE
    LET f = 4*r*x*(1-x)
  END IF
END

SUB bisection(r,p,xleft,xright,gleft,gright)
  DECLARE DEF f
  LET xmid = 0.5*(xleft + xright)    ! midpoint between xleft and xright
  LET gmid = f(xmid,r,p) - xmid
  IF gmid*gleft > 0 then
    LET xleft = xmid                 ! change xleft
    LET gleft = gmid
  ELSE
    LET xrigth = xmid                ! change xright
    LET gright = gmid
  END IF
END SUB
```

Check program period(p151, 2nd) or RecursiveFixedPointApp(p163, 3rd)

Root-Finding

- To find fix point, $x^* = f(x^*)$, one need to solve this equation, which is one of root-finding problems.
- Root-finding is commonly used in physics problems, e.g., mean-field approach.
- In the followings, we introduce a few methods of root-finding.

Root-Finding Methods

- Polynomial and Taylor Series
- Graphic Solution
- The Bisection Method
- Error Analysis
- The Newton-Raphson Method
- Rates of Convergence
- The Secant Method

Root-Finding Methods

- The False Position Method
- Hybridization of Bisection and Newton-Raphson or Secant Methods
- Iteration in General

See Root-Finding.pdf